

# Tracking the coffee: from production to the cup with Blockchain and IoT

*Treball Final de Grau*

Carles Juan Martínez

Director:  
Marc Pous Marín

Codirector:  
Jordi Paillissé  
Vilanova

Ponent:  
Josep Solé Pareta

25 de juny de 2019



---

## **Resum**

L'objectiu d'aquest projecte és el disseny i implementació d'un prototip del seguiment de la cadena de subministrament del cafè utilitzant Blockchain i Internet of Things per poder oferir al client final tota la informació de cada un dels passos de la línia de producció, des del començament fins a la tassa de cafè. El client hauria de tenir la certesa que el cafè és d'alta qualitat i que prové del camp, de l'agricultor i el país que el distribuïdor li diu. És un projecte fet a l'empresa thethings.io i vehiculat també amb el projecte europeu EPIC.

## **Abstract**

The goal of this project is the design and implementation of a prototype of tracking the coffee supply chain using Blockchain and Internet of Things in order to provide to the final client the information of each step of the production line, starting from the very beginning until the cup of espresso is on the table, ready to be tasted. The client should have the certainty that the coffee is a high quality coffee which comes from the field, from the manufacturer and from the country that the distributor says. The project is developed at thethings.io company and it is also done through the European EPIC project.

# Índex

Índex	iv
Índex de figures	vii
Índex de taules	viii
<b>1 Introducció i context</b>	<b>1</b>
1.1 Introducció . . . . .	1
1.2 Context . . . . .	2
1.3 Actors implicats . . . . .	2
1.3.1 Públic objectiu . . . . .	2
1.3.2 Qui ho usará . . . . .	2
1.3.3 Qui se'n beneficiarà . . . . .	3
1.4 Estat de l'art . . . . .	3
1.4.1 Blockchain . . . . .	3
1.4.2 Tipus de Blockchain . . . . .	5
1.4.3 Smart contracts . . . . .	6
1.4.4 Plataformes de Blockchain . . . . .	6
1.4.5 Projectes sobre Blockchain i cadena de subministrament . . . . .	8
1.4.6 Internet of Things . . . . .	8
1.4.7 Conclusió . . . . .	9
1.5 Formulació del problema i objectius . . . . .	10
1.5.1 Esquema general del projecte . . . . .	10
<b>2 Abast</b>	<b>12</b>
2.1 Objectius . . . . .	12
2.2 Possibles obstacles . . . . .	13
<b>3 Metodologia i rigor</b>	<b>14</b>
3.1 Metodologia . . . . .	14
3.2 Seguiment del projecte . . . . .	14
3.3 Eines de desenvolupament . . . . .	14

<b>4</b>	<b>Planificació temporal</b>	<b>16</b>
4.1	Descripció de les tasques . . . . .	16
4.1.1	Definició de les tasques i dependències . . . . .	16
4.1.2	Previsió temporal . . . . .	18
4.1.3	Recursos . . . . .	18
4.1.4	Diagrama de Gantt . . . . .	19
4.2	Valoració d'alternatives i pla d'acció . . . . .	22
<b>5</b>	<b>Gestió econòmica</b>	<b>23</b>
5.1	Identificació i estimació dels costos . . . . .	23
5.1.1	Recursos humans . . . . .	23
5.1.2	Recursos de software . . . . .	23
5.1.3	Costos dels recursos de hardware . . . . .	24
5.1.4	Costos indirectes . . . . .	24
5.1.5	Contingència i imprevistos . . . . .	25
5.2	Control de gestió . . . . .	27
<b>6</b>	<b>Hyperledger Fabric: Anàlisi</b>	<b>28</b>
6.1	Arquitectura . . . . .	29
6.2	Components d'Hyperledger Fabric . . . . .	30
6.2.1	Membership Service . . . . .	30
6.2.2	Ordering Service . . . . .	31
6.2.3	Gossip . . . . .	31
6.2.4	<i>Ledger</i> . . . . .	32
6.2.5	Channel . . . . .	32
<b>7</b>	<b>Disseny</b>	<b>33</b>
7.1	L'índex de qualitat del cafè . . . . .	33
7.2	Participants de la xarxa . . . . .	36
7.3	La xarxa Hyperledger Fabric . . . . .	37
7.3.1	Creant la xarxa . . . . .	37
7.3.2	El consorci . . . . .	38
7.3.3	L'element clau: El canal . . . . .	38
7.3.4	Peers, chaincodes i aplicacions . . . . .	38
7.3.5	Política d' <i>endorsement</i> . . . . .	39
7.3.6	Futura expansió . . . . .	39
7.4	El chaincode . . . . .	40
7.5	Els models . . . . .	42
7.6	L'API del sistema . . . . .	43
<b>8</b>	<b>Implementació del prototip</b>	<b>46</b>
8.1	Xarxa Hyperledger Fabric . . . . .	46
8.2	Chaincode . . . . .	49

## ÍNDIX

---

8.3	Verificació del chaincode . . . . .	56
8.4	Implementació de l'API . . . . .	57
<b>9</b>	<b>Verificació integral</b>	<b>60</b>
9.1	Connexió amb thethings.iO . . . . .	65
<b>10</b>	<b>Sostenibilitat i compromís social</b>	<b>68</b>
10.1	Dimensió econòmica . . . . .	68
10.2	Dimensió ambiental . . . . .	69
10.3	Dimensió social . . . . .	69
<b>11</b>	<b>Lleis i regulacions</b>	<b>71</b>
<b>12</b>	<b>Revisió dels compromisos inicials</b>	<b>73</b>
12.1	Planificació temporal . . . . .	73
12.2	Gestió econòmica . . . . .	73
<b>13</b>	<b>Conclusions</b>	<b>74</b>
13.1	Competències tècniques . . . . .	74
13.2	Altres competències . . . . .	75
13.3	Treball futur . . . . .	76
13.3.1	Mètriques de rendiment en casos reals . . . . .	76
13.3.2	Posada en producció: Kubernetes . . . . .	76
13.3.3	Dispositius IoT . . . . .	76
	<b>Bibliografia</b>	<b>79</b>
	<b>Annex</b>	<b>83</b>
<b>A</b>	<b>Aplicació mòbil</b>	<b>83</b>

# Índex de figures

1.1	Esquema del projecte . . . . .	11
4.1	Diagrama de Gantt . . . . .	21
6.1	Arquitectura execute-order-validate . . . . .	29
7.1	Pes de cada fase de la supply chain en la qualitat del cafè . . . . .	34
7.2	Esquema de la connectivitat dels participants i IoT amb el sistema . . . . .	37
7.3	Representació de la xarxa Blockchain . . . . .	40
7.4	Relació entre models . . . . .	43
8.1	Test mocha pel controlador de <i>cup</i> . . . . .	57
8.2	Arquitectura de l'aplicació . . . . .	58
9.1	World State de la Blockchain . . . . .	62
9.2	Creació del bloc gènesi . . . . .	63
9.3	Creació del bloc de <i>bag1</i> . . . . .	64
9.4	Ruta d'un sac de cafè . . . . .	67
9.5	Temperatura d'un sac de cafè . . . . .	67
A.1	Pantalla inicial i informació general . . . . .	83
A.2	Lectura d'identificadors de sacs . . . . .	84
A.3	Informació vària del seguiment del cafè . . . . .	84
A.4	Pantalla de donació a l'agricultor . . . . .	85

# Índex de taules

1.1	Comparativa de plataformes Blockchain . . . . .	7
4.1	Dedicació temporal . . . . .	19
4.2	Recursos de hardware, de software i humans . . . . .	20
5.1	Costos recursos humans . . . . .	24
5.2	Costos recursos de software . . . . .	25
5.3	Costos recursos de hardware . . . . .	26
5.4	Costos recursos indirectes . . . . .	26
5.5	Resum dels costos . . . . .	26
7.1	Paràmetres de qualitat del cafè . . . . .	35



# Capítol 1

## Introducció i context

### 1.1 Introducció

La necessitat de millorar les condicions de feina dels països exportadors de cafè combinat amb la gran demanda d'aquest producte arreu del món obre una gran oportunitat per treballar colze a colze amb la tecnologia. En el cas del cafè d'alta qualitat, la traçabilitat és un factor crític per assegurar-ne l'origen, la varietat i el procés aplicat [23]. Aquest escenari, en el qual la producció, la granja d'origen i el productor necessiten ser registrats, ens porta una gran oportunitat per aplicar la tecnologia per tal de millorar la confiança i la transparència entre consumidor, productor i intermediaris. L'objectiu del projecte, per tant, no és només implementar un sistema de traçabilitat del cafè al llarg de totes les etapes de la cadena de subministrament, sinó fer arribar tota aquesta informació al client final que beu la tassa de cafè al bar de forma verificada i transparent.

El projecte serà implementat a l'empresa *thethings.iO*, conjuntament amb l'empresa torradora de cafè –que per qüestions de confidencialitat serà anomenada d'ara en endavant *Cafès Barcelona*– i l'empresa intermediària que posa en contacte la torradora i el cultivador de cafè –en aquest cas, també serà mencionada amb el sobrenom de *Fair Trade Coffee*. En el cas que ens ocupa estem parlant d'un producte, el cafè d'alta qualitat, on l'agricultor controla tot el procés productiu fins a la seva exportació i on els controls de qualitat i la traçabilitat són essencials. La visibilitat del productor de cafè durant totes les etapes i el seu control del procés productiu garanteix que la major part de beneficis se'ls quedin precisament el productors.

Per tant, amb l'objectiu que l'empresa torradora aconseguixi fidelitzar els seus clients, es busca llançar una campanya de màrqueting a través de la qual connectar el client final amb l'agricultor per així aconseguir que el primer sigui capaç de conèixer i tenir a l'abast tots aquells factors que determinen la qualitat del cafè. Alhora, es pretén que la facilitació de tota aquesta informació reverteixi en unes millors condicions laborals per la persona que treballa el cafè [24]. Es pretén, a més, tancar el cicle i que l'agricultor pugui obtenir també informació per part del bevedor de cafè. El client, en posteriors etapes del projecte, hauria de ser capaç de retornar una recompensa cap al productor si així ho considera.

Aquest repte aporta l'oportunitat d'implementar una solució amb confluència de tecnologies àmpliament usades i en contínua exploració: Blockchain i smart contracts, conjuntament amb Internet of Things. En posteriors capítols s'explicarà el per què d'aquestes tecnologies com també l'abast d'aquest treball.

### 1.2 Context

Com s'ha dit, el treball es desenvoluparà a l'empresa thethings.iO, pel que el present treball és de modalitat B, segons estipula la normativa de Treball Final de Grau. Es desenvolupa també tenint en compte que l'especialitat de l'estudiant és la de Tecnologies de la Informació. thethings.iO és una empresa del sector de les TIC especialitzada en Internet of Things. Ofereix una plataforma a través de la qual connectar diversos tipus de dispositius i poder-ne visualitzar les dades a través de panells que l'usuari es pot fer a mida (estil *drag and drop*).

Finalment, per acabar de donar les claus del context, cal dir que el treball també se subscriu dins l'EPIC (EU Partnerships and International Cooperation), que forma part del programa Erasmus de la Unió Europea. Aquest programa posa en contacte diferents estudiants d'universitats de la Unió Europea i amb l'objectiu de treballar conjuntament en algun projecte ofert per alguna empresa. En aquest sentit, el projecte es divideix en diferents parts, de la qual el present TFG i la seva implementació n'és una.

### 1.3 Actors implicats

En aquest apartat s'explicaran els actors implicats o *stakeholders* del projecte.

#### 1.3.1 Públic objectiu

El producte està dirigit a dos actors. Un d'ells és el client, bevedor de cafè. Serà qui rebrà, en última instància, tota la informació de la cadena de subministrament, com també tots els paràmetres que determinen la qualitat del cafè. A través de tota aquesta informació, tant el productor, com Fair Trade Coffee, Cafès Barcelona i el bar on se serveixi el cafè poden garantir la qualitat del cafè. El client té la certesa, així, de que està pagant per un producte que val la pena.

En segon lloc, va dirigit al productor de cafè, qui rebrà la informació de retorn per part del client final i tindrà també coneixement de què li passa al seu cafè una vegada l'ha venut a l'empresa exportadora.

#### 1.3.2 Qui ho usará

Els actors que usaran el producte seran molts més que no només a qui va dirigit. Aquí ens referim no només al productor de cafè i al client final, ja explicats, sinó també a tots els agents involucrats dins la cadena de subministrament, com transportistes, treballadors de

l'empresa torradora, o qualsevol actor implicat en la cadena. Seran aquestes persones les que també aniran emplenant en diferents fases informació sobre la cadena, sempre i quan l'acció humana sigui requerida. Per exemple, entrega d'un carregament de cafè o dia en què es mol el gra.

### 1.3.3 Qui se'n beneficiarà

Amb aquest projecte se'n poden beneficiar totes les parts implicades. Tenir la informació de tota la cadena de subministrament –sobretot pel que fa a aspectes qualitatius del cafè– permet avaluar cada una de les fases per separat. Això vol dir que tots els agents són capaços d'avaluar cada una de les parts, alhora que poden ser avaluats per la resta. Fair Trade Coffee pot saber quins agricultors fan millor la seva feina, Cafès Barcelona pot saber si l'empresa transportista entrega quan toca els carregaments o si estan parats massa temps en un magatzem, de la mateixa manera que l'agricultor pot avaluar si val la pena vendre el seu cafè a Fair Trade Coffee, per posar només uns exemples.

A més, per l'empresa torradora és un projecte de màrqueting que pretén fidelitzar més clients amb els cafès d'autor i, per tant, augmentar els beneficis. Pels agricultors, la digitalització dels seus espais ha de permetre optimitzar les seves plantes, entendre què passarà amb el canvi climàtic, entendre en temps real la qualitat de forma tangible i fins i tot incrementar els seus preus d'acord a aquests factors.

## 1.4 Estat de l'art

Fins aquí s'han esmentat conceptes com Blockchain, Internet of Things, smart contract o cadena de subministrament. Anem a analitzar cadascun d'aquests conceptes i on es troben actualment aquestes tecnologies, com també projectes similars que pretenen o han implementat una solució semblant.

### 1.4.1 Blockchain

Partim de la base que, en el moment actual, no hi ha una base tecnològica que asseguri la confiança entre productor de cafè i client. Aleshores es requereix una tecnologia immutable on no es puguin desfer les escriptures que es facin, de tal manera que tothom tingui la certesa de que res no ha estat alterat. És en aquest context que Blockchain es presenta com a primera i única opció. La tecnologia Blockchain es defineix com 'un llibre de comptabilitat digital immutable distribuït i generalment sense una autoritat central' [34]. En altres termes, es pot dir que és usat per 'registrar transaccions entre diferents ordinadors i que aquestes no puguin ser alterades amb caràcter retroactiu sense l'alteració de tots els blocs de la xarxa i el protocol de consens de la xarxa entre els diferents nodes [36].

Malgrat que l'arquitectura del Blockchain depèn de la plataforma final que es fa servir sempre hi ha uns elements comuns en totes elles. A continuació es resumeixen [34]:

- Hash: El *hashing* és un mètode per calcular una sortida a partir d'una entrada determinada sempre amb una mateixa longitud. Qualsevol canvi, per petit que sigui, en l'entrada altera la seva sortida. Les tecnologies Blockchain creen *hashos* de totes les llistes de transaccions. Això significa que cada node que comparteix un mateix llibre de comptabilitat –d'ara en endavant *ledger* per usar la terminologia original– ha de poder generar el mateix hash.
- Transaccions: Una transacció és el registre d'una transferència d'un bé. Tant podria ser transferència de criptomoneda en el cas de Bitcoin o Ether, com transferència d'un quilo de cafè en el cas que ens ocupa.
- Criptografia asimètrica: La criptografia asimètrica usa un parell de claus interrelacionades, la clau pública i la clau privada. El parell de claus té múltiples funcions dins la tecnologia Blockchain. Les claus privades serveixen per signar les transaccions, mentre que les públiques per obtenir les adreces dels participants de la xarxa o per verificar signatures. El parell de claus també proveeix la capacitat de certificar que un participant que transfereix un bé a un altre realment està en possessió d'aquell bé.
- Adreces: Una adreça és una cadena de caràcters alfanumèrics que deriven de la clau pública més alguns camps de detecció d'errors. S'utilitzen per enviar i rebre béns.
- Ledger: O llibre de comptabilitat. Representa el conjunt de transaccions de la xarxa. En aquest sentit, cal un protocol de consens –quan s'expliqui la plataforma triada es farà més èmfasi en el protocol de consens en qüestió– per tal de verificar que totes les transaccions siguin vàlides entre els nodes de la xarxa.
- Blocs: Els usuaris envien les transaccions a un node de la xarxa. Aquestes transaccions esperen en una cua. Quan la cua ha arribat al nombre de transaccions que li pertoca (en funció de la plataforma) el node les publica a un bloc (conjunt de transaccions). Els conceptes en aquest sentit varien en funció també de la plataforma i en funció de l'algorisme de consens. Algunes plataformes com Bitcoin usen el concepte de minat, activitat per la qual determinats nodes descobreixen nous blocs a la xarxa on la resta de nodes poden anar-hi afegint conjunt de transaccions [29]. S'entrarà més en detall quan es parli de la plataforma seleccionada.

S'ha parlat en aquest punt també d'algorisme de consens mitjançant el qual els nodes es posen d'acord per actualitzar el *ledger*. És un mecanisme essencial, ja que tots els blocs de la xarxa van enllaçats en un ordre que no es pot canviar. Aleshores, de quina manera els nodes es posen d'acord en el següent bloc on escriure les dades? Això s'anomena mecanisme de consens. Els principals mecanismes són el *proof of work*, el *proof of stake* i *byzantine agreement*. Amb tot, en funció del tipus de Blockchain es poden desviar d'aquestes característiques.

El sistema de *proof of work* adopta una estratègia basada en incentius per encoratjar els nodes a participar en la xarxa. Els miners, o nodes que busquen nous blocs, proveeixen

recursos computacionals per descobrir nous blocs a canvi d'incentius [43] en forma de criptomoneda o tarifa sobre cada transacció afegida al bloc que han descobert.

En canvi, el *proof of stake* es basa en la idea de que com més presència tingui un node en la xarxa és més probable que el sistema tingui èxit i menys possibilitats que el node en qüestió vulgui comprometre el sistema. Aquest sistema es pot implementar de moltes maneres diferents, però en totes elles els nodes amb més presència són més proclius a generar nous blocs on gravar-hi les transaccions [34].

Per la seva banda, el *byzantine agreement*, usat per plataformes com Hyperledger Fabric (*Byzantine Fault Tolerance* o BFT), pretén gestionar les situacions en les quals, a causa d'un atac maliciós (generalment) algun o alguns nodes de la xarxa cauen per complet provocant que la informació que tenen la resta de nodes no sigui coherent [25]. Cal, doncs, un algorisme a través del qual posar-se d'acord i que la xarxa segueixi sent fiable. De fet, la tecnologia Blockchain es basa en els algorismes de consens per resoldre conflictes en entorns bizantins [35]. BFT seria un subtipus de *proof of stake*.

### 1.4.2 Tipus de Blockchain

Existeixen tres tipus de Blockchain diferents, les característiques de les quals són essencials a l'hora de definir el sistema. Existeixen les blockchain públiques, les privades i les mixtes, anomenades *permissioned*. A continuació una descripció d'elles [41]:

- Blockchain pública: Sorgeixen com a alternativa a l'administració central dels sistemes i és per això que són completament obertes. Qualsevol node es pot afegir a la xarxa i generalment hi ha incentius per unir-s'hi. Tots els participants poden llegir i escriure, sempre i quan se segueixin les normes de la xarxa. Bitcoin i Ethereum són les plataformes més conegudes en aquest tipus. Precisament per la seva definició, l'algorisme de consens que usen és el *proof of work*. Són adequades per solucions com el sistema de criptomonedes.
- Blockchain privada: Només s'hi pot accedir per invitació i tots els nodes són validats per l'organització que administra la xarxa o per les mateixes normes acordades entre els nodes participants. Són tipus de Blockchain adequades com a solucions internes de les empreses o organitzacions (regulació de contractes, empreses mèdiques, etc).
- Blockchain *permissioned*: També és coneguda com a blockchain de consorci o federada. En aquest cas, les transaccions són visibles només per aquells nodes que hi tenen els drets, no a tota la xarxa. L'algorisme de consens recau en alguns nodes predefinits. Es crea per tant una federació de nodes, atorgant certs drets a cadascun d'ells. És una solució molt adequada pels sistemes de cadenes de subministrament, on es busca que només certs participants puguin actualitzar certs béns (per exemple el seu propietari) però on tots els nodes que poden participar de la xarxa puguin fer-ne seguiment (lectures). Acaba sent un sistema que adopta propietats tant de les blockchain públiques com de les privades.

### 1.4.3 Smart contracts

El concepte d'*smart contract* o contracte intel·ligent ja va ser descrit pel precursor Nick Szabo el 1996 com “un conjunt de promeses, definides de forma digital, que inclouen protocols a través dels quals les diferents parts executen promeses” [38] [39]. Aquest seria el concepte que s'utilitzaria per després implementar-lo en plataformes com Ethereum. Amb tot, també és cert que el significat del concepte d'*smart contract* s'ha fet més ampli al llarg del temps, assumint una definició més genèrica com a “porció de codi que és desplegat en una xarxa Blockchain. Les transaccions enviades a la xarxa poden executar mètodes públics oferits per l'*smart contract*. El codi executa el mètode apropiat amb les dades proveïdes per l'usuari. El codi que es desplega i s'instancia a la xarxa és immutable i pot ser usat com un agent fiable de la xarxa” [34]. És per això que en funció de la plataforma aquest concepte pot tenir un nom o altre. És el cas d'Hyperledger on els contractes intel·ligents porten el nom de *chaincode*.

L'*smart contract* serà una peça fonamental del projecte a l'hora de definir la cadena de subministrament i els paràmetres de qualitat del cafè.

### 1.4.4 Plataformes de Blockchain

Arribats a aquest punt podem establir ja que la tipologia de Blockchain que s'ajusta més al projecte és una de *permissioned*. És per això que en aquest apartat es farà una comparativa de les principals plataformes d'aquest estil. Les que es compararan seran Ethereum (versió privada), Hyperledger Fabric i Corda. Deixem fora de l'abast d'aquest anàlisi plataformes públiques com Bitcoin ja que no són adients per al cas que ens ocupa (ja que són públiques). A continuació les principals característiques [40]:

Per una banda hi ha Fabric i Ethereum, flexibles però en diferents aspectes. Ethereum és molt potent a nivell d'*smart contracts*, però té un problema d'escalabilitat i privacitat. Fabric és més escalable gràcies al seu mode *permissioned* i al seu algorisme de tolerància a fallades bizantines (BFT). De totes maneres, Fabric té una estructura modular que permet adaptar-lo a multitud d'aplicacions diferents. En canvi Corda està creat per a pràcticament un sol cas d'ús: el món financer.

Fabric té a més un sistema molt potent de permisos a nivell d'atribut (ABAC) que el fa especialment ideal per al cas que ens ocupa. És cert que Ethereum és una plataforma àmpliament usada, però no s'ajusta tant al cas. És més, l'estructura de la xarxa de Fabric fa que es puguin assignar rols diferents a les organitzacions que participen de la xarxa, com també als usuaris que envien les transaccions, esdevenint així la plataforma triada per a aquest cas.

Fabric, a més, permet definir l'estructura de la xarxa aplicada al model de negoci. Com s'ha dit, thethings.io és l'empresa que desplega l'estructura de la xarxa Blockchain. El cas del cafè és només un cas d'ús concret i es llança com a pilot, però el sistema ha de poder ser escalable i aplicable a altres casos d'ús, com pugui ser el seguiment de la cadena de

Característica	Ethereum	Hyperledger Fabric	Corda
<b>Descripció</b>	Plataforma genèrica	Plataforma modular	Blockchain especialitzada en la indústria financera
<b>Governança</b>	Desenvolupadors d'Ethereum	Linux Foundation	R3
<b>Mode d'operació</b>	Pública o privada	Permissioned o privada	Permissioned o privada
<b>Consens</b>	Basada en mining i proof of work. A nivell de ledger	Permet diferents implementacions de consens. A nivell de transacció	Mecanisme de consens propi. A nivell de transacció
<b>Smart Contracts</b>	Smart contract code (Solidity)	Smart contract code (Go, Javascript, Typescript)	Smart contract code (Kotlin, Java) i Smart legal contract (termes legals)
<b>Moneda</b>	Ethers i tokens via smart contract	No té moneda ni té la necessitat de tenir-ne	No en té

Taula 1.1: Comparativa de plataformes Blockchain

producció del vi, d'alguna ginebra o del plàtan. Hyperledger permet la creació d'una sola xarxa però la possibilitat de crear canals interns independents de la resta. Els canals acaben actuant com una capa més, de manera que hi podria haver, dins la mateixa xarxa, un canal pel cafè, un pel vi, un per la ginebra i un altre pels plàtans, que no es podrien comunicar entre ells però sota la definició de xarxa de *thethings.iO*. Aquest és un dels arguments que també influeixen en la presa d'aquesta decisió. A més, en la versió 1.4 de Fabric es permet la creació de recursos privats que només poden ser compartits per certs nodes de la xarxa. Aquesta funcionalitat també li dona un gran potencial.

### 1.4.5 Projectes sobre Blockchain i cadena de subministrament

Establert l'estat de l'art pel que fa a la part tecnològica, es passa a repassar ara l'ús d'aquesta tecnologia en cadenes de subministrament i els principals projectes que existeixen.

No són ja poques les companyies que han aplicat Blockchain en la seva pròpia cadena de subministrament [21]. Amb tot, la majoria de projectes d'aquest estil no té com a finalitat el client final, com tampoc el productor, sinó més aviat l'eficiència dels processos empresarials de la pròpia empresa. Per tant, no tenen una dimensió pública, com sí la tendria el projecte que aquí s'exposa. És el cas d'*startups* com AxenS, Irene Energy o Zeto. Tampoc tenen l'objectiu de visibilitzar i dignificar la feina d'agricultors que, tradicionalment, han patit explotació laboral i on la indústria –en aquest cas la del cafè– ha generat un efecte devastador sobre el medi ambient.

Per tant, ens movem en uns àmbits on es mesclen les esferes empresarials (on és essencial i motora la voluntat de generació de benefici) però també esferes de caràcter ètic. És més, alguns autors apunten, i n'és precisament un dels objectius del projecte, que la tecnologia Blockchain pot ajudar els agricultors a obtenir beneficis més justos [24].

En aquest sentit sí que trobem, però, un cas semblant al projecte que es vol implementar. És el de *Provenance* [37], empresa que implementa tecnologia Blockchain a diferents casos de cadenes de subministrament. L'apliquen en casos d'ús com la pesca, la indústria del cotó o de la roba. I malgrat tenen el cas del cafè dins el seu *roadmap* no l'han implementat encara. Amb tot, *Provenance* el que persegueix és la traçabilitat via blockchain de la cadena de subministrament, sense entrar a valorar la qualitat del producte ni tampoc està centrat en el client final.

Troblem també algunes iniciatives, en estat molt embrionari, però que pretenen posar els primers grans de sorra amb projectes similars al que es presenta en aquest TFG. És el cas d'una *hackaton* a Costa Rica que tenia per objectiu la generació d'*smart contracts* per tal d'implementar el seguiment de la cadena de subministrament del cafè [42].

### 1.4.6 Internet of Things

El projecte necessita de la interacció amb persones físiques per tal d'anar emplenant els diferents factors de la qualitat del cafè, com més endavant està definit. Però no només les persones seran responsables de comunicar dades amb el sistema, sinó també dispositius connectats i diferents tipus de sensors.



En el món del cafè hi ha quatre casos d'ús dintre del món de l'Internet of Things (IoT) molt delimitats: el primer és el món agrícola, el segon la logística, el tercer la indústria i finalment la cafeteria. Cadascuna d'aquestes verticals tenen visibilitat avui en dia en l'IoT.

L'Internet of Things porta anys desenvolupant millores en l'àmbit de l'agricultura, tenint en compte la crisi climàtica que s'està vivint. Hi ha dos àmbits on especialment l'agricultura està millorant gràcies a la digitalització. El primer és l'agricultura de precisió i el segon és la robotització i automatització del camp amb empreses com John Deere [12] o Hello Tractor [9], a més de projectes de drons, entre d'altres. En quant a l'agricultura de precisió l'empresa catalana Blue Dots [18] és un referent. Aquesta última té com a objectiu obtenir la informació en temps real dels camps. Amb l'aplicació de mètodes d'intel·ligència artificial són capaços de predir la collita, instal·lant sensors prèviament el camp.

El segon punt que s'ha anomenat és la logística, un dels grans casos d'ús de l'IoT. El món del transport de mercaderies segueix funcionant igual des de fa 50 anys i la digitalització del mateix permet a clients i proveïdors poder entendre on són i l'estat de les mercaderies. Empreses com Sensolus [17] o Samsara [16] estan ajudant a digitalitzar el món de la logística.

La indústria i l'IoT, o altrament anomenat la indústria 4.0, ajuda a les empreses torradores a connectar la seva maquinària a Internet amb l'objectiu de poder predir aturades a la línia de producció, entre d'altres.

Finalment, l'IoT està entrant també en el món HORECA (hosteleria, restauració i cafeteries) per tal d'ajudar a les empreses a optimitzar, automatitzar i monitoritzar el què passa en els seus negocis. Les cafeteries, concretament, cada dia s'estan digitalitzant més, des de les geleries, tiradors de cervesa o cafeteres. Un dels primers exemples és la Nespresso Prodigio [15] que és la primera cafetera Nespresso connectada. Dit això, les cafeteres industrials ja comencen a estar connectades amb sistemes com el de Quality Espresso [2] on la cafetera, a part de monitoritzar-se, fa un manteniment predictiu d'ella mateixa amb l'objectiu d'oferir la millor qualitat de cafè durant el màxim de temps possible.

### 1.4.7 Conclusió

Revisada la bibliografia i havent fet un estudi dels actuals projectes existents al mercat podem concloure que Hyperledger és la tecnologia adequada per portar a terme el projecte. Té una gran comunitat al darrere i un ampli ventall d'eines que s'adapten perfectament al cas. A més, a nivell d'oportunitats, es conclou també que no hi ha cap altre producte en el mercat que estigui fent el es pretén aquí, pel que queda argumentada la inversió d'esforços en el projecte.

A més, l'estudi de l'estat de l'art conclou també que no hi ha alternatives tecnològiques a la proposta que es fa en aquestes pàgines, ja que a dia d'avui Blockchain és l'única tecnologia que ens permet garantir la immutabilitat de les dades.

Un últim punt a afegir és que com a element addicional als projectes estudiats, la proposta d'aquest TFG és fer ús de dispositius IoT per obtenir dades meteorològiques i GPS entre d'altres, de tal manera que tots els elements que intervenen en la qualitat del

café quedin perfectament registrats<sup>1</sup>.

### 1.5 Formulació del problema i objectius

Arribats en aquest punt tenim la informació suficient per a la definició del problema que es vol solucionar, a la vegada que els objectius. Assumim les premisses que la indústria del café ha tendit tradicionalment a generar relacions laborals pèssimes i a la destrucció del medi ambient. Veiem també com diverses empreses exportadores de café com també el moviment de comerç just els últims anys ha tendit a intentar revertir aquesta situació. Ens trobem doncs davant una necessitat i problema, que és: Com aplicar la tecnologia en aquest tipus d'indústria per tal de capgirar la situació?

**Objectiu 1:** Aplicar la tecnologia Blockchain a la cadena de subministrament del café per a generar una millora en les condicions dels productors de café.

Un element essencial respecte l'objectiu 1 és el client final. Cal que el consumidor sigui conscient d'on prové el café que beu i sota quines condicions. Això propiciarà o augmentarà les probabilitats que vulgui pagar-ne el preu just.

**Objectiu 2:** Fer arribar tota la informació de la cadena de subministrament de forma immutable i transparent a cadascun dels actors implicats fins a arribar al client final.

Un tercer element de la formulació del problema és un que s'ha anat repetint al llarg d'aquestes pàgines. Com demostrar la qualitat del café? Caldrà quantificar, per tant, els paràmetres que incideixen directament en la qualitat final del café, que comencen en la sembra de l'arbre del café i acaba en la forma en com es prepara la tassa del café en el bar.

**Objectiu 3:** Generar un índex de qualitat que permeti avaluar el producte final consumit: La tassa de café.

#### 1.5.1 Esquema general del projecte

Establerts els objectius podríem fer l'esquema complet del projecte (figura 1.1, font pròpia).

La part que fa referència al present TFG és la part de sobre de l'esquema. És a dir, la connectivitat amb la xarxa Blockchain, a través de l'actor o dispositiu IoT que sigui, en cada una de les fases de la cadena de subministrament. Es pot observar, també, que el consumidor retorna a la granja una recompensa en forma de moneda electrònica. Com s'ha dit a l'inici, aquesta part s'ha d'explorar en profunditat en futures etapes del projecte, però no pas en el TFG. En tot cas, es contempla com una part essencial del projecte.

---

<sup>1</sup>Aquests paràmetres de qualitat provenen d'entrevistes directes amb agricultors del país d'origen, com també de la bibliografia (Brita Folmer, 2017).

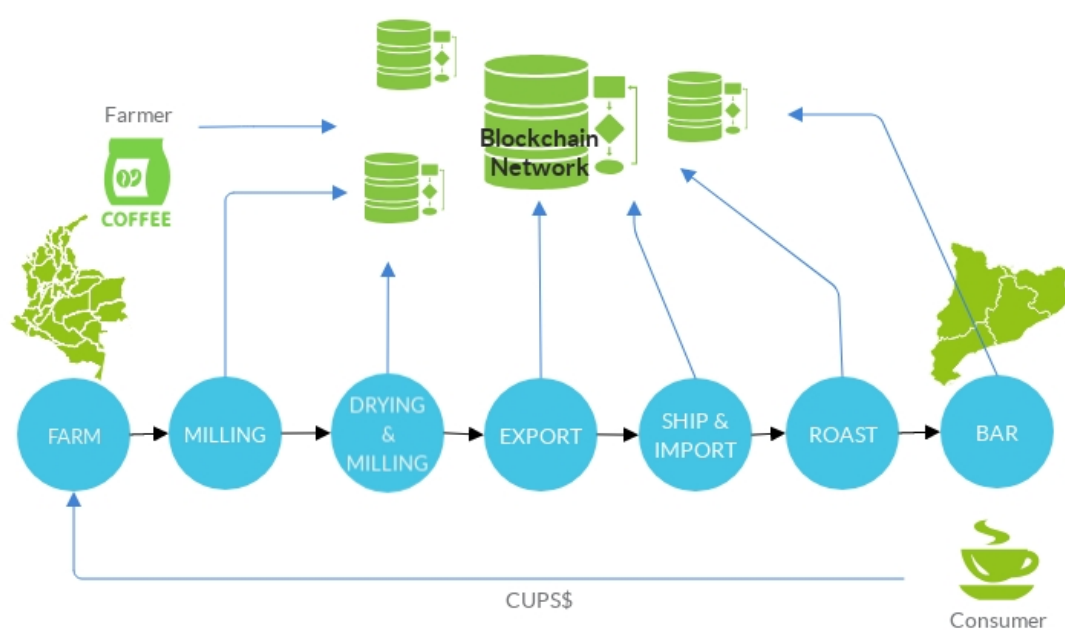


Figura 1.1: Esquema del projecte

# Capítol 2

## Abast

El projecte definit fins al moment és un projecte de llarg abast i que es podria dividir en diverses tasques grans:

- Disseny i implementació de la xarxa Blockchain i *smart contracts*, així com la generació d'una API per interaccionar amb la xarxa. Com s'ha dit, el projecte també se subscriu en el programa EPIC. Aquesta correspondria a una de les parts i la que es desenvoluparà en aquest treball.
- Implementació del sistema de retorn en forma de recompensa (mencionat a la introducció).
- Implementació de les apps d'usuari, sigui a través d'una app mòbil o des del navegador. Aquesta és l'altra que es fa a través de l'EPIC. Part desenvolupada per un estudiant de l'Estat turc.

### 2.1 Objectius

L'objectiu d'aquest TFG és centrar-se en el primer punt que s'ha esmentat. És a dir, en el disseny i la implementació de la xarxa Blockchain. Per això, l'objectiu final del TFG és aconseguir les següents fites:

- Disseny dels *smart contracts* –a partir d'ara *chaincode* per usar els termes d'Hyperledger Fabric–.
- Disseny de com haurà de ser la xarxa Hyperledger real i la seva API.
- La implementació d'un entorn local que emuli exactament el cas d'ús amb el desplegament del *chaincode* que posteriorment serà portat a producció.
- Implementació d'una API a través de la qual interaccionar amb la xarxa local.

## 2.2 Possibles obstacles

Els principals obstacles que podrien sorgir són:

- Limitació temporal: La durada en la qual s'ha de presentar el TFG és un temps curt. A vegades els temps venen també determinats per l'empresa i els diferents agents implicats. Això implica que cal ser estricte i constant.
- Hyperledger Fabric és un sistema molt nou, malgrat que té una gran comunitat al darrere. Està en constant evolució. Això implica que van sorgint noves versions, com també nous *frameworks* per treballar. Implica també que la documentació pot no ser del tot exhaustiva en segons quins aspectes. Tot això pot fer que exigeixi més dedicació que en d'altres tecnologies.

# Capítol 3

## Metodologia i rigor

### 3.1 Metodologia

La metodologia usada en aquest projecte serà en cascada. Aquesta és una metodologia seqüencial en la qual es passa per les etapes de requeriments, disseny, implementació, verificació i manteniment [22]. Cal tenir en compte que el temps per portar a terme el TFG és molt curt i és portat a terme per una sola persona, pel que la fase de manteniment no es farà. Les alternatives, com les metodologies *agile* i iteratives són més complicades d'implementar. No obstant, al ser una prova pilot per l'empresa i, en cas d'obtenir resultats positius, el que és ara un projecte en cascada podria esdevenir la primera iteració d'una metodologia iterativa, incloent noves persones al projecte. En tot cas, això sí que queda fora de l'abast del projecte i del temps del TFG.

### 3.2 Seguiment del projecte

El projecte és revisat permanentment (a diari) pel director del TFG, atès que és de modalitat empresa. Així mateix, s'han establert reunions cada dues setmanes amb el ponent i codirector (també de la Facultat d'Informàtica de Barcelona) per tal de fer-ne el seguiment.

### 3.3 Eines de desenvolupament

Les eines de desenvolupament seran les següents:

- Els binaris d'Hyperledger Fabric 1.4, que en el dia d'escriptura de la memòria –primavera 2019– és la darrera versió.
- Els *frameworks* utilitzats per generar l'entorn de desenvolupament són Convector i Hurley, que permeten les proves de *chaincodes* per tal de depurar tots els errors.
- A nivell empresarial, s'utilitza bitbucket com a repositori pel control de versions.

- Per a la planificació de tasques i objectius s'utilitza l'eina online TeamGantt.
- S'utilitzarà Overleaf (antic sharelatex) per a la redacció de tots els documents.

# Capítol 4

## Planificació temporal

### 4.1 Descripció de les tasques

En aquest apartat es definiran totes les tasques del projecte, com la seva duració, recursos associats, dependències entre elles i, finalment, el diagrama de Gantt. Hem de tenir en compte que el Treball Final de Grau té una càrrega de 18 crèdits ECTS, és a dir, de 450 hores<sup>2</sup>. Tot i així, la normativa de la UPC contempla que excepcionalment un crèdit pot equivaldre a 30 hores en casos de TFG o pràctiques externes. S'ajustarà a la temporalitat mínima

#### 4.1.1 Definició de les tasques i dependències

Les tasques es divideixen en cinc grans grups. En cada un dels apartats s'explicaran les subtasques corresponents, la seva duració associada i les dependències. Posteriorment s'agruparà en un quadre resum i en el Gantt es veurà la seva distribució en el temps i les dependències gràficament.

#### Gestió de Projectes

La part de Gestió de Projectes (GEP) fa referència a la primera etapa del TFG. El mòdul consta de 3 crèdits ECTS, que equivalen a 75 hores de càrrega de treball. Les hores estan repartides de la següent manera:

1. Abast del projecte i contextualització: 24,5 hores.
2. Planificació temporal: 8,25 hores.
3. Gestió econòmica i sostenibilitat: 9,25 hores.
4. Presentació oral i document final: 18,25 hores.

---

<sup>2</sup>En el moment d'escriure aquestes línies la data de defensa pública del TFG encara no està tancada, pel que s'assumirà que serà el primer dia dels possibles (1 de juliol).



5. Preparació presentació oral i presentació oral: 14,75 hores.

Cadascuna d'aquestes fases són dependents una de l'altra en l'ordre en què estan posades i no es poden començar sense haver acabat l'anterior.

### Fase de disseny

La fase de disseny és fonamental per encarrilar la fase d'implementació de l'entorn de proves. Tindrà una duració total de 70 hores. La fase de disseny és dependent de la de Gestió de Projectes excepte en la tasca Preparació presentació oral i presentació oral. Les tasques que la conformen són:

1. Definir l'índex de qualitat de cafè: Cal definir els paràmetres que influeixen en la qualitat del cafè. Aquests paràmetres hauran d'estar presents dins el *chaincode*. Es farà en base a la bibliografia sobre cafè i entrevistes a grangers i Fair Trade Coffee.
2. Definició de models: Definir els participants que poden interaccionar amb el *chaincode* i amb quins privilegis.
3. Disseny del *chaincode* i API: Fase crítica del projecte, doncs el *chaincode* és la columna vertebral del projecte. És l'element amb el qual interaccionaran tots els participants. També cal definir la API per tal que les aplicacions que vulguin fer crides a la Blockchain sàpiguen quin seran els endpoints i el format de les dades.
4. Disseny de la xarxa Blockchain. Malgrat que l'entorn serà local, es dissenyarà la xarxa real que s'implementarà en un futur. Per tant, cal pensar de quins nodes i organitzacions estarà conformada i quines seran les polítiques que la regiran.

Les tasques 1 i 2 són independents entre elles i les dues depenen de la fase de GEP. La tasca 3 depèn de 1 i 2 i la tasca 4 és independent de 1, 2 i 3.

### Implementació de l'entorn de proves

Tercera fase del projecte. Malgrat ser tan important com la resta serà la que requerirà més temps. És per això que es preveu que compregui 165 hores del total. Estarà composta per les següents tasques:

1. Adaptació a Hyperledger Fabric i Convector: Es preveuen un seguit d'hores per tal d'adaptar-se a la nova tecnologia més a fons, aprendre sobre models reals i fer totes les proves necessàries amb l'entorn.
2. Creació del *chaincode*: Si una de les fases crítiques era el disseny del *chaincode*, també ho serà la seva implementació. És la tasca on s'hi dedicarà més temps.
3. Emulació de la xarxa Blockchain: Caldrà emular la xarxa on el *chaincode* ha de ser instanciat.

4. Creació de la API: Implementació de la API que interaccionarà amb la Blockchain.

La tasca 1 precedeix a la 2, 3 i 4 en qualsevol cas i depèn de que estigui acabada la fase de disseny. La 2 i 3 són independents entre elles i la 4 depèn de la 2.

### Testing

Fase final d'implementació en la qual es provaran tots els components i el seu correcte funcionament. Les hores dedicades seran 80:

1. Testing pròpiament dit i les possibles modificacions que s'hagin de fer si es troben errors. Es faran tests sobre el *chaincode* (50 hores).
2. Testing d'integració amb la xarxa blockchain (30 hores).

La tasca 1 depèn de la 4 de l'anterior apartat, mentre que la 2 depèn tant de la 4 com de 3 de l'apartat anterior.

### Documentació final

Aquesta serà l'última etapa de projecte i ocuparà les darreres 60 hores. Les tasques seran:

1. Redacció de la memòria final: S'haurà de plasmar tota la feina feta des de la fita inicial i els resultats obtinguts, pel que les hores dedicades a aquesta seran considerables, tot i que comprimides en poc temps.
2. Preparació de la defensa pública: Inclou l'elaboració de la presentació per projectar com també la posada en escena i la defensa de tot el projecte.

La tasca 1 depèn de tota la fase d'implementació, mentre que la tasca 2 depèn que s'hagi acabat d'escriure la 1.

### 4.1.2 Previsió temporal

La taula 4.1 relaciona cada una de les tasques explicades en l'apartat anterior amb la seva durada temporal.

### 4.1.3 Recursos

En la taula 4.2 es detallen els recursos que es necessitaran pel projecte.

Tasca	Temps
<b>GEP</b>	<b>75 hores</b>
Abast del projecte i contextualització	24.5h
Planificació temporal	8.25h
Gestió econòmica i sostenibilitat	9.25h
Presentació oral i document final	18.25h
Preparació presentació i presentació	14.75h
<b>Fase de disseny</b>	<b>70 hores</b>
Definició índex qualitat del cafè	15h
Definició de models	15h
Disseny chaincode i API	30h
Disseny xarxa blockchain	10h
<b>Implementació</b>	<b>165 hores</b>
Adaptació Hyperledger i Convector	30h
Chaincode	60h
Emulació xarxa Blockchain	45h
API	30h
<b>Testing</b>	<b>80 hores</b>
Chaincode	50h
Integració chaincode amb Hyperledger	30h
<b>Documentació final</b>	<b>60 hores</b>
Redacció de la memòria final	50h
Preparació defensa pública del TFG	10h
<b>TOTAL</b>	<b>450 hores</b>

Taula 4.1: Dedicació temporal

#### 4.1.4 Diagrama de Gantt

La figura 4.1 representa el diagrama de Gantt on es detallen totes les tasques que s'han exposat en aquest capítol, repartides en el temps i amb les dependències explicitades gràficament.

Recurs	Explicació
<b>Hardware</b>	
Ordinador de sobretaula HP Intel Core I5-2400 Quad Core, 16GB RAM	Ordinador de l'empresa TheThings.iO
Monitor Acer K242Hl de 27"	Monitor de l'empresa TheThings.iO
Ordinador portàtil Dell XPS15	Ordinador personal.
<b>Software</b>	
Sistema operatiu: Linux Ubuntu 18.04	Sistema operatiu de l'empresa.
Control de versions: Bitbucket	Control de versions i servidor remot.
Visual Studio Code	IDE
TeamGantt	Eina online per planificar tasques.
Programari entorn de treball: Binaris de Hyperledger Fabric, framework Convecton i Hurley.	Proveït per Linux Foundation i Worldsibu.
Editors: LibreOffice i OverLeaf (antic ShareLatex)	Per elaborar els documents.
draw.io	Eina online per dissenyar esquemes i gràfics.
<b>Humans</b>	
Director, codirector i ponent i professor de GEP.	Avaluació treball i consells per millora i guies per a la implementació (en funció del perfil)
Desenvolupadors de TheThings.iO	Suport en temes puntuals
Grangers de cafè, contacte amb Fair Trade Coffee i Cafès Barcelona	Ajuda en chaincode i índex qualitat.
Estudiant (autor del TFG)	Autor del TFG, executarà totes les fases.

Taula 4.2: Recursos de hardware, de software i humans

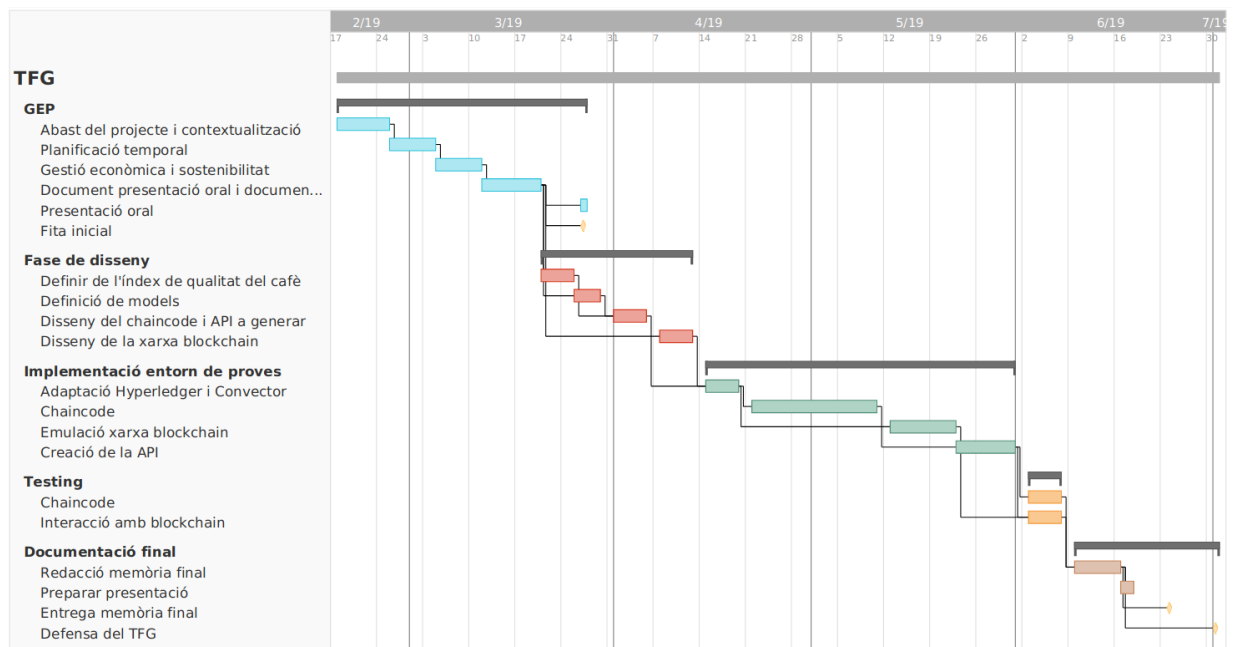


Figura 4.1: Diagrama de Gantt

## 4.2 Valoració d'alternatives i pla d'acció

En aquest apartat cal, primer, identificar els riscos que formen part del projecte. En la fase de GEP es considera que pràcticament no hi ha riscos. Si bé és cert que es tracta d'un mes amb entregues setmanals hi ha un baix risc de desviacions. Per tant, no es proposa pla d'acció.

En la fase de disseny es contempla un risc de dilatació de la fase perquè el *chaincode* i l'índex de qualitat del cafè no només es fa en base a bibliografia, sinó en base a converses i entrevistes amb agricultors, Fair Trade Coffee i Cafès Barcelona. Els grangers i Fair Trade Coffee són d'altres països amb zones horàries diferents. L'obtenció de la informació es podria demorar per aquesta qüestió.

Per atacar aquests possibles riscos es garanteix que el director del projecte estarà personalment implicat pel que fa a la concertació d'aquestes entrevistes o xerrades. Es valora qualsevol tipus de mitjà per facilitar-ho, ja sigui videoconferència com correu electrònic.

La tercera fase, la d'implementació de l'entorn de proves, és la més llarga i és on hi ha més riscos, de caràcter tècnic. Al ser la fase que més hores aglutina és clarament el coll d'ampolla del projecte en cas que alguna cosa falli. A més, a *thethings.io* no hi ha cap expert en Blockchain al ser ara mateix un projecte d'I+D+i.

Atès que la majoria de riscos provenen del domini de la tecnologia (binaris, *framework*,...) i com es diu cap altre desenvolupador de l'empresa podrà ajudar en aquest sentit es valora la possibilitat de tenir una ajuda directe per part dels desenvolupadors del *framework*. Ja s'ha fet el contacte i la voluntat d'aportar ajuda en cas de problemes s'ha constatat.

En la fase de testing es contempla un risc obvi, i és que en aquesta fase es descobreixin que hi ha més errors del que es pensava.

Tot i que és un risc obvi, s'ha estat generós en el temps dedicat a aquesta fase a fi de poder assumir qualsevol de les modificacions que s'hagin de fer en aquesta fase. Per aquest motiu no es preveuen més accions per tal d'atacar el risc, que ja ha estat contemplat.

L'última fase, la de documentació final, en si mateixa no té risc intrínsec més que la conseqüència d'allargar alguna de les fases anteriors i que, per tant, no hi hagi el temps suficient per elaborar la memòria correctament. Igual que en la fase de testing, aquest risc es vol prevenir a priori, és per això que s'han dotat a totes les fases d'hores extra en concepte de possible dilatació en el temps a causa dels riscos anomenats anteriorment.

# Capítol 5

## Gestió econòmica

En aquest capítol es tractarà la dimensió econòmica de la matriu de sostenibilitat des de la perspectiva del pressupost.

### 5.1 Identificació i estimació dels costos

La identificació dels costos es farà en base a la descripció dels recursos de la planificació temporal. Es relacionarà també la dedicació del recurs en funció de l'activitat en el diagrama de Gantt.

#### 5.1.1 Recursos humans

A la taula 5.1 es presenten els costos referents als recursos humans.

Com es pot apreciar, no s'han comptabilitzat econòmicament les hores que fan referència al codirector, al ponent, a l'agricultor, i als CEO de Fair Trade Coffee i Cafès Barcelona. Pel que fa als dos primers, formen part de la Facultat d'Informàtica de Barcelona de la UPC, pel que el seu sou és assumit per la institució i no per thethings.io. Passa el mateix amb l'agricultor i els CEO.

#### 5.1.2 Recursos de software

A la taula 5.2 es passen a detallar els costos relacionats amb els recursos de software.

Com es pot veure, els costos referents als recursos de software són nuls. S'ha intentat al màxim optar per plataformes de codi obert que, si bé no tenen per què ser gratuïtes, generalment ho són.

Recurs humà	Dedicació	Fase(s)	Preu/Hora	Cost (€)	total
Director del projecte	50	Totes	100	5.000	
Codirector del projecte	30	Totes	-	-	
Ponent del projecte	30	Totes	-	-	
Desenvolupador TheThing.iO (suport)	20	Implementació (Disseny Chaincode)	20	400	
Agricultor	2	Disseny (Definició índex qualitat)	-	-	
CEO Fair Trade Coffee	5	Disseny (Definició índex qualitat)	-	-	
CEO Cafès Barcelona	10	Disseny (Definició índex qualitat)	-	-	
Estudiant	450	Totes	9	4.050	
<b>Total</b>	<b>592</b>	-	-	<b>9.450 €</b>	

Taula 5.1: Costos recursos humans

### 5.1.3 Costos dels recursos de hardware

En la taula 5.3 es detallen els costos dels recursos de hardware:

### 5.1.4 Costos indirectes

El treball es desenvoluparà a les oficines de thethings.iO i això implica que hi haurà una sèrie de despeses indirectes. Es comptabilitza el lloguer de l'espai i les despeses d'oficina. En les despeses d'oficina es tenen en compte aspectes com la neteja del local, Internet, etc. Veure la taula 5.4 per a més informació. Cal tenir en compte que el projecte no assumeix la totalitat de despeses com el lloguer, ja que s'ha de comptabilitzar la part proporcional. El que es fa és dividir el preu del lloguer (o despeses) pel nombre de treballadors i en relació al percentatge de la jornada laboral de l'estudiant dedicada al TFG. Les hores dedicades al TFG suposen, més o menys, un 40% de la jornada laboral de l'estudiant i a l'oficina hi treballen físicament 3 treballadors.



Recurs software	Preu	Unitats	Vida útil	Amortització (€/h)
Linux Ubuntu 18.04	0	2	3 anys	-
Bitbucket	0	1	-	-
Visual Studio Code	0	1	-	-
TeamGantt	0	1	-	-
Binaris Hyperledger, Convector i Hurley	0	1	-	-
LibreOffice	0	1	-	-
Overleaf	0	1	-	-
draw.io	0	1	-	-
<b>Total</b>	<b>0</b>	-	-	-

Taula 5.2: Costos recursos de software

### 5.1.5 Contingència i imprevistos

Pel que fa al càlcul d'imprevistos i contingències cal dir que el projecte no té cap fase concreta en la qual les despeses siguin majors a la resta. És a dir, que els costos són constants al llarg dels 5 mesos de duració del TFG. Tenint en compte que el projecte s'entregarà a finals de juny, la dilatació d'una o altra fase en detriment d'alguna altra no suposarà un increment de les despeses.

El que sí caldria seria afegir una despesa de contingència, entesa com una partida per atenuar errors d'informació o descuits. Afectaria sobretot a la fase d'entrevistes amb agricultors i empreses intermediàries en la fase de disseny.

En tot cas, sí que en l'apartat de planificació temporal es va preveure que precisament la fase de disseny es pogués allargar més del compte i que s'hi dedicarien més recursos si fes falta. Això implica possibles desplaçaments per reunions extra. També en la fase d'implementació, al ser la més llarga, es contemplava ajuda extra d'altres desenvolupadors de l'empresa. I això implicarien més hores de la seva jornada laboral dedicades al projecte. En aquest sentit, parlariem d'un augment d'hores per part del director i de desenvolupadors (la resta d'actors no tenen repercussió sobre el pressupost). Es preveu que es puguin necessitar 30 hores més per part del director durant la fase de disseny i 15 hores més de desenvolupadors. És a dir, de 1200€ extra per part del director i de 300€ de desenvolupadors.

Amb aquesta exposició, es considera un nivell de contingència del 7,5% (inclou els imprevistos del paràgraf anterior).

Recurs hardware	Preu	Unitats	Vida útil	Hores	Amort. (€/h)	Total (€)
Ordinador de sobretaula HP Intel Core I5-2400 Quad Core, 16GB RAM	305	1	4 anys	760 hores	0,04	30,4
Monitor Acer K242Hl de 27"	114	1	4 anys	760 hores	0,01	7,6
Ordinador portàtil Dell XPS15	1455	1	4 anys	200 hores	0,18	36
<b>Total</b>	<b>1874</b>	-	-	-	<b>0,23</b>	<b>74€</b>

Taula 5.3: Costos recursos de hardware

Recurs indirecte	Preu	Temps	Cost (€)
Lloguer de l'oficina	700€ / 3 * 0,4	5 mesos	467
Despeses de l'oficina	1000€ / 3 * 0,4	5 mesos	667
Transport	20€	5 mesos	100
<b>Total</b>	-	-	<b>1.234€</b>

Taula 5.4: Costos recursos indirectes

Recurs	Pressupostat (€)
Recursos humans	9.450
Recursos de software	0
Recursos de hardware	74
Costos indirectes	1.234
Contingència i imprevistos	1.500
<b>Total</b>	<b>12.258€</b>

Taula 5.5: Resum dels costos

## **5.2 Control de gestió**

El control de gestió és essencial per tot projecte. En l'apartat anterior s'ha previst que la màxima desviació possible del pressupost sigui del 7,5%. Això implica establir mecanismes de control del pressupost i de les conseqüències econòmiques de qualsevol desviació.

Per tal de garantir que qualsevol desviació es combat amb suficient temps es faran reunions setmanals amb el director del projecte per avaluar la consecució de les fites i valorar els problemes que hagin pogut sorgir (més enllà del seguiment diari que s'ha mencionat anteriorment).

Així mateix, i com ja s'ha dit en anteriors apartats d'aquest document, també es faran reunions mensuals amb ponent i codirector, de perfil acadèmic, per valorar també i abordar qualsevol tipus de desviació que afecti a les seves competències.

## Capítol 6

# Hyperledger Fabric: Anàlisi

Tot i que en l'apartat d'estat de l'art s'ha descrit Hyperledger Fabric s'ha volgut dedicar un capítol sencer del TFG a analitzar i justificar la decisió de triar aquesta plataforma per implementar la xarxa Blockchain del projecte. És important entrar en el detall per, posteriorment, en els capítols de disseny i implementació entendre les decisions que es prendran.

Hyperledger Fabric és un projecte de codi obert iniciat per la *Linux Foundation*. Com s'ha dit anteriorment, una xarxa Blockchain és un llibre de comptabilitat immutable on s'hi fan transaccions, mantingut per una xarxa distribuïda de nodes. Són aquests nodes que mantenen còpies aplicant les transaccions (agrupades en blocs enllaçats entre ells) que han estat validades mitjançant un protocol de consens. En el cas que ens ocupa, a més, volíem una plataforma Blockchain on els participants fossin identificables, que la xarxa fos *permissioned*, que tengués un alt rendiment en les transaccions, baixa latència i la capacitat de ser una xarxa confidencial i privada. Segons els seus creadors, Hyperledger Fabric es va dissenyar seguint aquestes premisses en un context d'ús empresarial [32].

La plataforma ha estat dissenyada des del seu nucli per adaptar-se a les necessitats empresarials de cada cas. S'anomenaran ara els mòduls, tot i que després s'entrarà en detall de què significa cadascun d'ells:

- Mòdul per l'*ordering service*, que estableix el consens en l'ordre de les transaccions i després fa *broadcast* dels blocs als *peers*.
- El mòdul del *membership service provider* (MSP), responsable d'associar cada una de les entitats de la xarxa amb el material criptogràfic.
- Un mòdul opcional *peer-to-peer gossip service* per difondre els blocs a altres *peers*.
- Un element que marca la diferència també amb altres plataformes és el mòdul pels *chaincodes*, que és executat en un contenidor docker a part. El codi, així, pot ser escrit en diferents llenguatges de programació. En aquest mateix mòdul es poden triar diferents *Data Base Management System* (DBMS).

- Finalment, el mòdul de la política d'aprovació (*endorsement*) i validació, que es pot configurar de forma independent per a cada aplicació.

Quan es diu que Hyperledger és *permissioned*, i com s'ha dit també a l'estat de l'art, significa que opera sobre un grup de participants coneguts, identificats i (en algunes ocasions) vetats operant amb un model de governança que implica un cert grau de confiança. Aquest tipus de Blockchain proveeix una forma d'assegurar les interaccions entre aquests usuaris que, entre si, no tendrien per què tenir confiança mútua. Confiant en les identitats dels participants, una Blockchain *permissioned* pot utilitzar protocols de *crash fault tolerance* (CFT) o de *byzantine fault tolerance* (BFT) que no requereixen cap cost de minat.

## 6.1 Arquitectura

Hyperledger Fabric es basa en l'arquitectura *execute-order-validate*, que es pot resumir en la gràfica següent [20]:

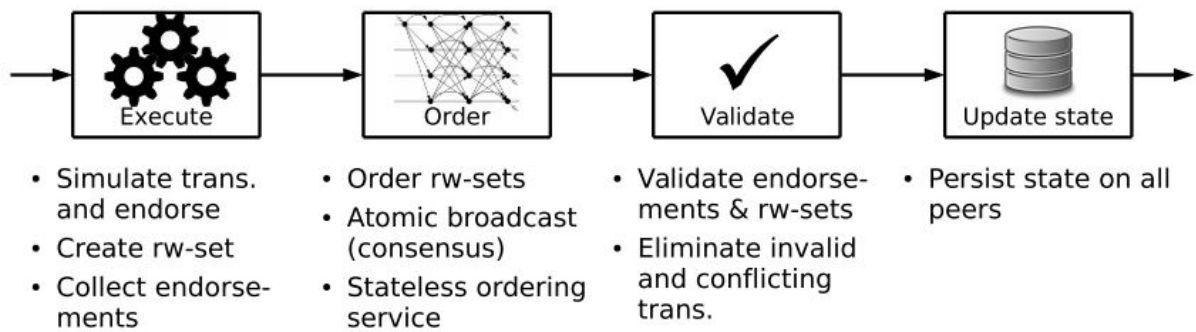


Figura 6.1: Arquitectura execute-order-validate

Resumidament, una aplicació distribuïda per Fabric consisteix en dues parts:

- El *chaincode*: Més concretament, un *chaincode* és el codi que implementa la lògica de l'aplicació i corre durant la fase d'execució. És la part central d'una aplicació distribuïda en Fabric i pot ser escrit per un desenvolupador no conegut (*untrusted*). També hi ha *chaincodes* especials que serveixen per gestionar la xarxa i són anomenats de sistema.
- Política d'aprovació: És evaluada durant la fase de validació. Les polítiques d'aprovació no poden ser modificades ni triades per desenvolupadors no coneguts pel sistema. Actua com una llibreria estàtica per a la validació de transaccions. Només els administradors de la xarxa poden tenir permisos per fer-hi modificacions. Una política d'aprovació típica permet els *chaincodes* especificar els nodes *endorsers* per a una transacció. Utilitza una lògica de l'estil de "tres de cinc" o " $(A \wedge B) \vee C$ ".

Un client envia una transacció als *peers* especificats per la política d'aprovació. Cada transacció és executada per cert *peers* i el seu resultat guardat. Aquesta és el pas d'aprovació. Després de l'execució, les transaccions entren en la fase 2 del gràfic, Order, que consisteix en executar el protocol de consens i produir una seqüència ordenada de transaccions anomenada bloc. Llavors es fa *broadcast* d'aquestes seqüències a tots els *peers*. Posteriorment s'entra en la fase 3, de validació, on cada peer valida les transaccions en el mateix ordre i de forma determinista.

Com que Hyperledger Fabric és una xarxa *permissioned* tots els nodes tenen una identitat proveïda pel *Membership Service Provider* (MSP). Així, els nodes poden assumir un d'aquests tres rols:

- Client: Envien propostes de transaccions, ajuden a disposar la fase d'execució i, finalment, fan *broadcast* de les transaccions.
- Peer: Executen les propostes de transaccions i les validen. Tots els *peers* mantenen el *ledger*, una estructura de dades on només s'hi pot afegir informació, guardant la cadena de hashos, així com l'estat (representació de l'últim estat). És el que s'anomena rol de *commiteeing*. Però no tots els *peers* executen totes les propostes de transaccions. Els que sí que ho fan són anomenats *endorsing peers* i és un rol reservat a aquells que tenen instal·lat el *chaincode*. A més, però, hi ha el rol de líder que és aquell node que distribueix les transaccions entre els nodes d'una mateixa organització i que pertanyen a un mateix canal.
- Ordering Service Node: Són els nodes que conjuntament conformen l'*Ordering Service*. Resumidament, s'encarreguen d'establir l'ordre entre totes les transaccions de la xarxa, on cada transacció conté actualitzacions i dependències computades durant la fase d'execució, juntament amb les signatures criptogràfiques de cada *endorsing peer*. Els *orderers* no són conscients de l'estat de l'aplicació i no participen en l'execució ni validació de transaccions. Això permet que el protocol de consens a Fabric sigui modular i fàcilment intercanviable en funció del cas d'ús de l'empresa o organització.

## 6.2 Components d'Hyperledger Fabric

A continuació es passen a detallar els components que s'usaran de Hyperledger en el present projecte en base a les definicions de [20] i [32].

### 6.2.1 Membership Service

L'MSP manté les identitats de tots els nodes de la xarxa (clients, *peers* i OSN) i és responsable d'expedir noves credencials que són usades per a l'autenticació i autorització. Com que Hyperledger Fabric és *permissioned* totes les interaccions que es donen dins la xarxa entre els nodes es fa a través de missatges que són autènticats amb signatures digitals. L'MSP inclou un component a cada node des d'on pot autènticar les transaccions, signar i

validar i autenticar també altres operacions Blockchain. Les eines per la gestió de les claus i registre també formen part de l'MSP.

La implementació per defecte de l'MSP suporta mètodes estàndard de PKI per autenticació basats en signatures digitals i poden allotjar Autoritats de Certificació (CA) comercials. Amb tot, Hyperledger implementa una CA independent anomenada Fabric-CA.

Un concepte molt important i que es repetirà al llarg d'aquestes pàgines és el d'organització, definit com a grup de membres del Blockchain. En Hyperledger cada organització gestiona el seu propi MSP. És a dir, que quan una organització és convidada a participar d'una xarxa Fabric té la capacitat per incloure-hi els seus propis participants expedint certificats digitals amb el seu propi MSP. Si una organització participa de diversos canals (explicats a continuació) en una mateixa xarxa podria tenir un MSP per a cada un d'ells.

### 6.2.2 Ordering Service

L'Ordering Service gestiona diferents canals (més endavant s'expliquen el que són). En cada canal proveeix diferents serveis:

- *Atomic broadcast* per establir l'ordre de les transaccions, implementa el *broadcast* i serveix les diferents crides.
- Reconfigura els canals quan un dels seus membres modifica la configuració.
- Opcionalment, implementa el control d'accés

L'Ordering Service és arrencat amb un bloc gènesi en el canal de sistema. Aquest bloc porta en una transacció la configuració de les propietats del mateix Ordering Service.

Podem trobar dues implementacions principals de l'Ordering Service: Kafka i Solo. La implementació utilitzada en producció és una instància d'Apache Kafka [1]. Kafka ofereix una solució escalable i que aguanta fallades dels nodes basada en ZooKeeper. Kafka pot executar-se en un node a part de l'Ordering Service, qui actua com a proxy cap als *peers*. La implementació anomenada Solo és per entorns de desenvolupament i és centralitzada.

### 6.2.3 Gossip

Hyperledger optimitza el funcionament de la xarxa Blockchain, la seguretat i l'escalabilitat dividint la càrrega de treball entre els nodes encarregats d'*endorsing* i *commiting* i els *ordering* nodes. Aquest desacoplament de les operacions de xarxa requereix un protocol que asseuri la integritat i consistència de les dades. És el que s'anomena *Gossip data dissemination protocol*.

Les funcions del protocol són:

1. Gestionar el descobriment de *peers* i els membres d'un canal.
2. Disseminar les dades del *ledger* a través de tots els *peers* de la xarxa.

3. Posar al dia ràpidament els nous *peers* permetent la transferència *peer-to-peer* de les dades.

En aquest sentit, hi ha un tipus de node, anomenat *anchor* que són utilitzats per aquest protocol per assegurar que els *peers* de diferents organitzacions es coneixen els uns als altres.

### 6.2.4 *Ledger*

Un llibre de comptabilitat és el component que s'encarrega de mantenir l'estat actual d'una empresa i del registre de transaccions que s'han fet fins al moment. El *ledger* en Fabric té una funcionalitat idèntica. És a dir, traduït a aquesta tecnologia significa que té dues funcions bàsiques diferenciades però interrelacionades: Manté el *world state* (la fotografia actual, implementat amb CouchDB) i el Blockchain (el registre de transaccions passades). Segons els documents oficials de Fabric [32], el *world state* és útil pels programadors ja que és de lectura ràpida i és una base de dades que només manté l'últim estat de la xarxa (de la manera en què l'última transacció l'ha deixat) en forma de clau-valor.

La Blockchain és el registre de totes les transaccions que donen com a resultat el *world state*. Les transaccions, com s'ha explicat en l'apartat d'estat de l'art es recopilen en blocs i són afegides al Blockchain. Aquesta és la base de dades que és immutable i no pot ser alterada retrospectivament.

### 6.2.5 *Channel*

Finalment, un element essencial per entendre Hyperledger Fabric i que també és molt important per entendre la implementació que se'n farà per satisfer el model de negoci són els canals. La base de Fabric són els canals, entesos com una subxarxa privada de comunicació entre dos o més membres. Com a mínim sempre hi haurà d'haver un canal. Un canal és definit pels seus membres, els *anchor*, el *ledger*, el *chaincode* i l'*order service node*.

La importància dels canals és molt gran ja que permet que una sola xarxa Blockchain pugui satisfer el model de negoci d'una empresa. En el nostre cas, podem entendre que l'empresa thethings.iO serà l'encarregada de configurar la xarxa i crear els canals. En el cas d'ús del cafè, aquest serà un canal on hi participaran l'empresa mateixa, l'empresa torradora de cafè i l'empresa exportadora. Paral·lelament, però, thethings.iO podria tenir un altre client que vol aplicar Blockchain a la cadena de subministrament del vi. Així doncs, només caldria crear un canal nou on es convidessin els membres i organitzacions involucrades en el cas. Els canals del cafè i del vi, com a norma general i per defecte, no es poden comunicar entre si i resten aïllats. Això permet tenir una sola estructura. Cada canal estaria regit pel seu propi o propis *chaincodes*.



# Capítol 7

## Disseny

L'apartat anterior posa sobre la taula els conceptes i els elements d'Hyperledger Fabric necessaris per fer el disseny del projecte en aquest apartat.

El disseny que es presenta en aquestes pàgines és la posada en producció del projecte. No obstant, en l'apartat d'implementació es farà una emulació en entorn local, un prototip. Això sí, el codi que s'executarà serà exactament el mateix en ambdós casos (*chaincode*) i la implementació d'aquest prototip servirà per validar el correcte funcionament en un entorn controlat.

### 7.1 L'índex de qualitat del cafè

Comencem primer per la part del disseny que no és estrictament tècnica. Com s'ha dit al llarg d'aquesta memòria, un dels objectius és aportar tota la informació relativa a la cadena de subministrament del cafè, però a més, incloure els paràmetres que incideixen directament amb la qualitat del cafè. D'aquesta manera el client final tindrà a la mà aquells factors que han incidit directament en la producció de la tassa de cafè. La taula 7.1 es deriva d'entrevistes amb agricultors, exportadors de cafè i torradors de cafè per així determinar els factors clau.

Un cop tenim els paràmetres de qualitat del cafè ja tenim els elements que cal anar recopilant en el *chaincode* i que seran introduïts pels diferents participants de la Blockchain, ja sigui l'agricultor, com un transportista, l'empresa torradora o el barista. També els dispositius IoT tendran la funció d'emplenar variables com les condicions climàtiques o la geoposició de la granja o dels paquets de cafè al llarg de la seva ruta. En el següent subapartat s'especificaran els diferents tipus d'actors que poden interaccionar amb la xarxa Blockchain.

Com també es pot apreciar a la taula, cada una de les variables té un pes sobre el total. Aquest percentatge ha de servir per poder donar un índex de qualitat final i que servirà de base al client consumidor per tenir una idea general de la qualitat del cafè. Malgrat que igualment es donarà la informació de cada un dels passos, el client no té per què saber si una temperatura de 18°C a una altura de 1800 metres i al voltant de l'equador és

acceptable o no per a la qualitat del cafè. És per aquest motiu que es vol donar un resultat final al client.

A continuació es presenta un gràfic resum de cada una de les fases i el seu percentatge en relació a la qualitat del cafè (figura 7.1). A més, establerts els objectius podríem fer l'esquema complet del projecte (taula 1.1).

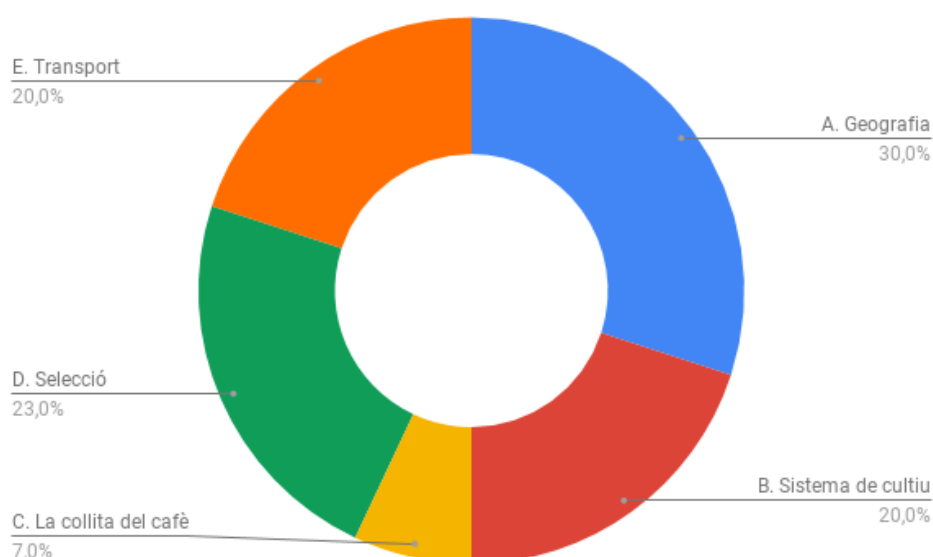


Figura 7.1: Pes de cada fase de la supply chain en la qualitat del cafè

---

<sup>3</sup>Les entrevistes s'han fet amb agricultors llatinoamericans i no s'ha trobat el terme traduït al català. *Beneficio* fa referència al procés de despolpat, fermentació i rentat de la cirera del cafè.

<sup>4</sup>El gra pergamí és el gra que s'obté després de l'assecat i l'envolta una petita capa protectora anomenada pergamí. En aquest moment s'eliminen els grans defectuosos.

<b>A</b>	<b>Ubicació geogràfica</b>	<b>30%</b>
1	Altitud del terreny	7%
2	Latitud del terreny	5%
3	Condicions climàtiques específiques	6%
4	Amplitud tèrmica diària	7%
5	Característiques del sòl	5%
<b>B</b>	<b>El sistema de cultiu</b>	<b>20%</b>
6	Varietat cultivada	6%
7	Nutrició del cultiu	3%
8	Densitat i orientació de la sembra	2%
9	Ombres associades al cultiu	2%
10	Gestió de malalties del cultiu	2%
11	Gestió de plagues	2%
12	Cultiu de plantes acompanyants	1%
13	Edat del cultiu	2%
<b>C</b>	<b>La collita del cafè</b>	<b>7%</b>
14	Recol·lecció selectiva de les cireres	6%
15	Aplec, transport i manipulació de les cireres	1%
<b>D</b>	<b>El <i>beneficio</i><sup>3</sup> humit, sec i la trilla</b>	<b>23%</b>
16	Classificació, selecció i avaluació de les cireres	3%
17	Separació de la polpa	1%
18	Selecció del gra en mucílag	1%
19	Fermentació del gra	5%
20	Rentat del gra	3%
21	Selecció del gra rentat	1%
22	L'assecat del gra	4%
23	Selecció del gra pergami <sup>4</sup>	1%
24	Emmagatzematge del gra pergami	1%
25	La trilla del gra	1%
26	Selecció del cafè d'alta qualitat	1%
27	Emmagatzematge del cafè d'alta qualitat	1%
<b>E</b>	<b>Torrat, processament i preparació de la beguda</b>	<b>20%</b>
28	Transport del cafè	1%
29	Torrat del cafè d'alta qualitat	7%
30	Envasat del producte	1%
31	Distribució del producte	1%
32	Moldre el cafè torrat	4%
33	Preparació i servei de la beguda	6%

Taula 7.1: Paràmetres de qualitat del cafè

## 7.2 Participants de la xarxa

Entenem com a participant tota persona que pugui interaccionar amb la xarxa, però també dispositius IoT. En aquest sentit, hi participaran les següents entitats o persones físiques:

- Agricultor del cafè: Interacciona amb el sistema amb una aplicació mòbil amb vistes específiques pels agricultors.
- Treballador de l'empresa exportadora (els quals assumeixen tasques com l'envasament o enviament): Igual que l'agricultor interacciona amb app mòbil amb vistes específiques.
- Transportistes: Interacciona amb el sistema amb una app mòbil on només podrà inserir informació relativa a la recollida i entrega de carregaments.
- Treballadors de l'empresa distribuïdora: Aquí hi tenen cabuda activitats com l'emmagatzematge del cafè un cop arriba a Europa, la seva gestió i la distribució entre les diferents empreses torradores. Interaccionen amb el sistema de forma anàloga a com ho fan els treballadors de l'empresa exportadora.
- Treballadors de l'empresa torradora: El mateix que el punt anterior respecte a la interacció amb el sistema.
- Client final: Interacciona amb el sistema amb una app mòbil on podrà consultar tota la informació de la cadena de subministrament. Tot i que no entra dins l'abast del TFG, eventualment podria ser escanejant un codi QR que estés imprès al sobre de cafè.

Pel que fa a dispositius IoT, es conceben els següents:

- Estació meteorològica que pugui capturar dades de geolocalització, temperatura, humitat, nivell de pluja, entre d'altres.
- Dispositiu de geolocalització per GPS, per fer el seguiment dels sacs entre la seva sortida del camp on han estat cultivats fins que arriben al bar.
- Màquina de cafè. Màquina amb diferents sensors encarregats de comptabilitzar, entre d'altres, dades com la temperatura de la beguda, els mil·lilitres d'aigua subministrats, els grams de cafè servits o el tipus de cafè (expresso, americano, machiato,...). D'aquesta manera, al bar, no hi hauria la necessitat de que cap treballador interactués amb el sistema a l'hora de la preparació de la tassa de cafè.

Havent repassat els diferents tipus de participants i dispositius IoT podríem resumir la seva interacció amb el sistema amb la figura 7.2 (font pròpia).

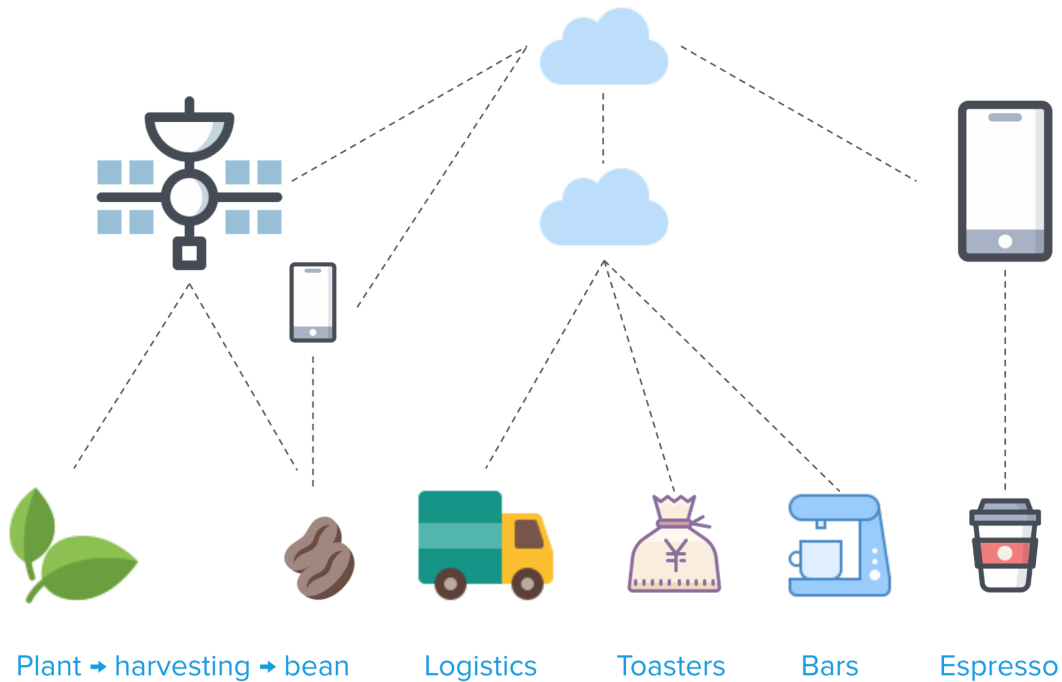


Figura 7.2: Esquema de la connectivitat dels participants i IoT amb el sistema

## 7.3 La xarxa Hyperledger Fabric

S'usarà l'esquema 7.3 per anar explicant la configuració de la xarxa Hyperledger real en base als conceptes de l'apartat 6.

### 7.3.1 Creant la xarxa

La xarxa es crea quan s'afegeix el primer node d'*Ordering Service*  $ON_{TTIO}$  i és el primer element que la defineix. En aquest cas, crearem la xarxa de *thethings.iO*.  $ON_{TTIO}$  és configurat i iniciat per un administrador de l'organització *thethings.iO*  $O_{TTIO}$ . La configuració  $NC_{TTIO}$  conté les polítiques que descriuen el conjunt de drets que s'atorguen inicialment a una organització. Inicialment aquests drets s'atorguen només a  $O_{TTIO}$ . Tot i que en un futur es podrien afegir més administradors a la xarxa, el cas d'ús que ens ocupa no contempla més administradors de xarxa.

També es pot apreciar l'Autoritat de Certificació  $CA_{TTIO}$ , que serveix, com s'ha dit, per expedir certificats als diferents nodes i participants. Els certificats expedits serviran als *endorsers* per signar les transaccions i donar-hi validesa. La configuració de xarxa  $NC_{TTIO}$  (així com la resta de forma anàloga) usa un MSP per identificar la propietat d'un certificat dispensat per  $CA_{TTIO}$  i que associa un propietari amb l'organització  $O_{TTIO}$ .

### 7.3.2 El consorci

Un cop hi ha una primera xarxa muntada cal definir un consorci  $X1$  per  $O_{TTIO}$ , únic amb drets per fer-ho. Un consorci és un grup d'organitzacions amb uns objectius comuns i que tenen una necessitat de fer transaccions entre ells. El consorci serà la base per, posteriorment, crear un canal. Parlem, doncs, del consorci del cafè. S'afegeixen així les organitzacions de Cafès Barcelona  $O_{CB}$  i Fair Trade Coffee  $O_{FTC}$ .

També tenim les autoritats de certificació  $CA_{CB}$  i  $CA_{FTC}$ . Com a norma, cada organització de la xarxa sol tenir les seves pròpies autoritats de certificació per a identificar els seus propis membres i nodes.

Inicialment es contempla que tots els participants de la xarxa formaran part d'una d'aquestes tres organitzacions. Els agricultors seran participants que inclourà Fair Trade Coffee, els treballadors dins l'empresa torradora seran participants creats per Cafès Barcelona, així com el bar on es consumeixi el cafè. La resta de participants com transportistes o dispositius IoT seran participants que inicialment crearà l'organització thethings.iO. En un futur podria ser que s'afegís alguna altra organització interessada en cobrir algun d'aquests perfils.

### 7.3.3 L'element clau: El canal

Posteriorment s'afegeix a la xarxa un canal  $C_{Coffee}$ . El canal és la via de comunicació pels membres d'un consorci. Tal i com s'ha dit anteriorment, un canal és una via de comunicació privada. El canal és creat per les organitzacions  $O_{TTIO}$ ,  $O_{CB}$  i  $O_{FTC}$ . El canal està subjecte a la configuració de  $CC_{Coffee}$  que està separada de la  $NC_{TTIO}$ . Cal notar, doncs, que al ser configuracions separades, els drets que té l'organització thethings.iO en el canal són els mateixos que Cafès Barcelona i Fair Trade Coffee. Es podria haver donat el cas que thethings.iO hagués creat el consorci sense voler-ne ser membre, un cas que se sol donar. En aquest cas només tindria drets per eliminar el consorci, però no per saber què hi passa dins, doncs és una via de comunicació privada entre les organitzacions que en formen part. Si a més  $O_{TTIO}$  crea el consorci i dona drets als membres que hi formen part de poder crear canals, podran fer-ho perfectament. És aquí on Hyperledger mostra un dels seus grans potencials al tenir diferents capes de configuració que permeten la segmentació i adaptació de la xarxa a l'estructura de negoci. Es podria crear un nou consorci també, com s'ha dit anteriorment, per un altre producte, com el vi, amb unes altres organitzacions dins la mateixa arquitectura.

### 7.3.4 Peers, chaincodes i aplicacions

Fins aquí la configuració està preparada, però encara no hi ha l'habilitat d'interaccionar ni de fer transaccions. Falten una sèrie d'elements que inclourem ara. En primera instància, cada organització haurà d'afegir un *peer node*  $P_{TTIO}$ ,  $P_{CB}$  i  $P_{FTC}$  (tenen els drets per a fer-ho). Cada un d'aquests nodes seran els encarregats de mantenir una còpia del ledger  $L_{Coffee}$  corresponent al canal  $C_{Coffee}$ . A la vegada, cada un d'aquests nodes també tindrà

associada una identitat, expedida per la corresponent Autoritat de Certificació de cada organització.

Ara serà el moment d'instal·lar i instanciar el *chaincode*  $S_{Coffee}$  (en properes seccions s'explicarà el seu disseny) per un administrador. La instal·lació només es farà en el peer de *thethings.iO*. En segon lloc, l'administrador de *thethings.iO* haurà d'instanciar-lo a  $C_{Coffee}$  per a que la resta de peers de les altres organitzacions en tinguin coneixement. Tot i que tots els nodes podran usar ara el *chaincode* només aquells on hagi estat instal·lat podran veure'n el codi.

Finalment s'afegeix una aplicació per a cada organització  $A_{TTIO}$ ,  $A_{CB}$  i  $A_{FTC}$ . Aquesta aplicació, en el nostre cas seria en forma d'API (explicada també en apartats posteriors), està associada a cada una de les organitzacions i tot i que està fora de la xarxa Blockchain hi està connectada a través del canal del cafè i es regeix pel *chaincode*. Així doncs, els usuaris finals usaran aquestes aplicacions que invocaran el *chaincode* i d'aquesta manera enviaran propostes de transaccions als *peers*.

### 7.3.5 Política d'*endorsement*

La política d'*endorsement* o d'aprovació és, com ja s'ha dit, un element cabdal. Està definida a  $CC_{Coffee}$ , pel que totes les organitzacions en són conscients. En el nostre cas serà del tipus (AND(TTIO.peer, FTC.peer, CB.peer)), és a dir, la transacció serà validada sempre i quan un mínim d'1 peer de cada organització ho faci.

### 7.3.6 Futura expansió

El que s'ha vist fins ara és només una simplificació de la xarxa per explicar-ne el seu disseny. Veiem que l'Ordering Node  $ON_{TTIO}$  és el punt central de la xarxa i només n'hi ha un. Si caigués cauria tota la xarxa. Per solucionar això cal afegir uns quants Ordering Nodes extres. Com a mínim dos més. En cas que la xarxa general fos administrada també per altres organitzacions, aquestes podrien posar els seus propis nodes Ordering. Com s'ha dit també s'hauria d'usar el mètode *Kafka-Zookeeper*, ja que el *solo* és només per entorns de desenvolupament.

Finalment, també cal mencionar que només hi ha un *peer* per organització per a simplificar-ho. Però l'ideal seria que cada una de les organitzacions pogués tenir més d'un node. Al tenir diferents nodes poden assumir així diferents rols (que ja s'han explicat a 6.1).





- Participant de la xarxa: Pot ser un participant de carn i ossos, com els agricultors, transportistes o baristes, o bé pot ser un dispositiu d'Internet of Things, com un sensor de temperatura o un dispositiu GPS.
- Camp o granja: És el primer punt de la cadena de subministrament. El camp és on es cultiven els arbres del cafè i d'on surt el gra en brut. Les primeres variables que influeixen en la qualitat del cafè seran en el camp. De la taula 7.1, els apartats A, B, C i part de D és informació que es recull al camp. És a dir, més del 50% de la qualitat. Hi intervenen dispositius IoT i l'agricultor.
- Sac de cafè: És l'agricultor l'encarregat d'elaborar els primers sacs de cafè provinent de tot allò que ha recollit. El sac és una bossa gran de cafè i el nombre de quilos pot variar. Cada sac serà identificable i només portarà cafè d'un únic agricultor. Seran aquests sacs que s'enviaran i arribaran a Europa gestionat per Fair Trade Coffee, qui ho vendrà a Cafès Barcelona. Actualitzaran informació d'aquesta estructura a la Blockchain, per tant, l'agricultor, dispositius IoT (bàsicament pel seguiment GPS) i transportistes. En aquest sentit, cal veure que un camp de cafè tindrà com a "fills" X sacs de cafè.
- Bossa de cafè: Una vegada Cafès Barcelona rep el sac de cafè el comença a processar i inicia el procés de torrat. Una vegada fet els empaqueta en bosses petites per a vendre als bars. Igual que fins ara, aquestes bosses són íntegrament amb el cafè d'un sol agricultor i mai es mescla. Un sac de cafè tindrà alhora com a "fills" X bosses de cafè. Actualitzaran informació d'aquesta estructura de dades Cafès Barcelona i els transportistes que portaran les bosses als bars.
- Tassa de cafè: Estructura de dades final de la cadena de subministrament. Cada un dels objectes d'aquest tipus serà "fill" d'una de les bosses de cafè. D'aquesta manera es completa la cadena de subministrament. Actualitzaran aquesta estructura només els baristes i la màquina de cafè (dispositiu IoT).

El model de participant segueix el model *Attribute-Based Access Control* (ABAC) basat en els patrons per gestionar les identitats a Hyperledger Fabric [33]. ABAC és un mètode on els participants que sol·liciten fer operacions sobre objectes són autoritzats o vetats a fer-ho en funció dels atributs del participant en qüestió, atributs assignats a l'objecte, condicions externes o un conjunt de polítiques [31].

Passem a comentar, doncs, la lògica de negoci. En aquest sentit, els participants de la xarxa només podran fer modificacions sobre aquells objectes dels quals en siguin propietaris en aquell moment. Això significa que un agricultor no ha de poder modificar atributs de com es fa la tassa de cafè, de la mateixa manera que un barista no podrà, sota cap circumstància, afegir temperatures a l'*array* d'amplitud tèrmica. Cal tenir en compte que quan l'agricultor cedeixi X sacs de cafè a Fair Trade Coffee ja no tindrà els permisos per a fer-hi escriptures. En quant als permisos de lectura, qualsevol participant de la xarxa podrà consultar qualsevol informació. En el cas dels dispositius IoT serà diferent. En

aquest cas un objecte podrà tenir assignats un o varis dispositius/participants els quals podran actualitzar informació sobre la geolocalització, la temperatura, la humitat, etc.

La lògica de negoci en relació als diferents objectes del cafè és la següent. Només un participant amb perfil 'agricultor' podrà crear un objecte de *camp de cafè*. Serà l'agricultor i un o més dispositius IoT que aniran emplenant l'estructura de dades relativa als factors de qualitat del cafè. Un cop el procés hagi acabat l'agricultor inserirà el total de quilos de cafè i crearà tants objectes *sac de cafè* com pugui. Quan en creï una el codi verificarà que se li passa una ID de camp de cafè correcte, que n'és el propietari i que pot crear el sac. L'objecte *camp de cafè* quedarà tancat un cop no quedin quilos i ja no es podrà modificar.

Quan es crea un objecte *sac de cafè*, de la mateixa manera que passa amb les bosses i tasses, hereta l'estructura de dades relativa als paràmetres de qualitat del cafè. D'aquesta manera cada objecte té tot el recorregut anterior. El que s'ha explicat fins aquí és també anàleg per la resta d'objectes

D'aquesta manera el client final serà capaç de veure qui li ha produït el cafè, quants quilos i sota quines circumstàncies es van fer, cap on van anar els sacs i posteriorment les bosses.

## 7.5 Els models

Com s'ha dit, els models de la Blockchain seran el participant, el camp de cafè, el sac, la bossa i la tassa de cafè. A continuació es detallen els atributs de cada un dels models.

- Participant: Els atributs d'un participant seran el seu nom, el Membership Service Provider al qual pertany (en altres paraules, la seva organització), un array d'identitats i el rol. L'array d'identitats emmagatzemarà les identitats x509 de l'usuari i fa referència a que un participant pot canviar de certificat digital i això no ha de comportar un canvi d'usuari. Cada participant a la xarxa hauria de poder canviar de certificat. Aquest canvi, però, només podrà ser autoritzat per un administrador de la xarxa.
- Crop: Objecte referent al camp de cafè. Els seus atributs seran la data de creació, usuari que el va crear, data de modificació, usuari que modifica, el seu propietari, granja, país, geolocalització, número de quilos recol·lectats i número de quilos restants (d'on es van restant quan es crea un sac de cafè) i un booleà per saber si l'objecte està tancat i per tant si es pot modificar.
- Sack: Objecte referent al sac de cafè. Es crea a partir d'un objecte de crop. Els seus atributs són, a més de les dades de creació i modificació (comunes a la resta d'objectes), la ID del crop del qual deriva, la capacitat del sac, els quilos de cafè i el booleà per saber si està l'objecte tancat. També conté un array de transferències. És a dir, el sac és el primer objecte que pot passar d'un usuari a un altre (d'agricultor a transportista, per exemple) i cal portar un registre de per quines mans ha passat aquell sac.

- Bag: Objecte referent a la bossa de cafè. A nivell de model (que no de controlador) és molt similar al sac.
- Cup: Objecte referent a la tassa de cafè que es crea i es consumeix al mateix bar. Té els atributs assignats a sack i bag i a més podria incloure, si així ho accepta, la informació de l'usuari que l'ha consumit.

Cal dir que les estructures anteriors estan relacionades entre elles 1 — N. És a dir, hi ha un camp que pot tenir N sacs de cafè i així successivament. Per aquest motiu, els fills hereten dels pares diverses estructures. Una és l'estructura de les variables qualitatives del cafè. Una altra és l'arxiu de transferències i finalment la ruta amb les geoposicions per així tenir la ruta que ha seguit. L'estructura de transferències no és més que la ID de l'usuari origen, el de destí i la ID de l'objecte transferit. L'estructura de les geoposicions és un array on es guarden geoposicions i el seu timestamp. Finalment, l'estructura de les variables qualitatives és l'estructura que implementa les dades explicades a 7.1. S'explicarà més en detall a l'apartat d'implementació. A més, com s'ha dit anteriorment, cada un dels objectes tindrà també un array de dispositius IoT que tenen permisos per a fer-hi escriptures (però que no s'heretarà).

La següent figura il·lustra gràficament la relació entre els models:

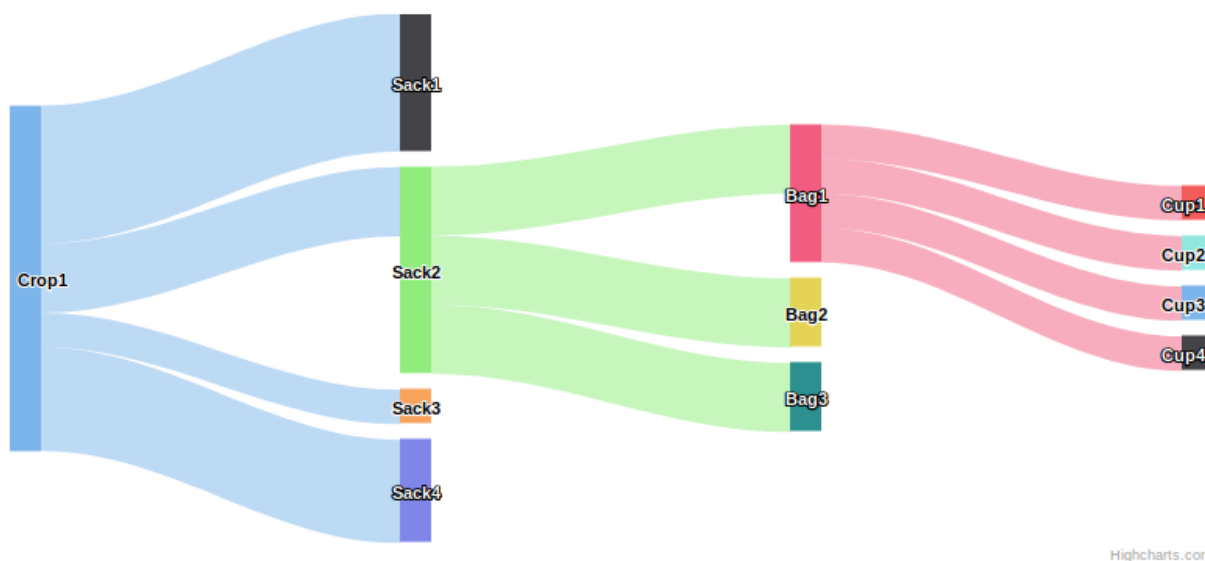


Figura 7.4: Relació entre models

## 7.6 L'API del sistema

Un dels últims punts a tractar en l'apartat de disseny és el de l'API. La manera en què les aplicacions d'usuari podran interactuar amb la Blockchain és a través de l'API, que estarà

encarregada d'atacar la xarxa gràcies al *chaincode*. Els controladors dels models seran els encarregats de subministrar les dades que l'API necessita o de sol·licitar les transaccions.

Inicialment es dissenya una API senzilla a través de la qual poder fer transaccions a la Blockchain. És per això que bàsicament inclourà la creació de noves estructures de dades (participant, camp, sac, bossa i tassa de cafè), l'obtenció d'informació i l'inserció de les variables de la qualitat del cafè. Tractarem per una banda el participant i per l'altra les estructures que fan referència al cafè, que a nivell d'API tendran les mateixes crides.

- POST */participant*: Hi haurà un mètode POST per crear els participants. Cal tenir en compte que tota interacció a la Blockchain es fa identificant-se a través d'un certificat digital. Això significa que tot usuari anirà lligat a un certificat. D'aquesta manera podem implementar ABAC a nivell de codi en el *framework* Convector. Inicialment només caldrà passar com a paràmetres el nom i el rol del participant. Precisament pel que es deia anteriorment, no és necessari contrasenya, ja que un cop creat el participant i es lligui a un certificat digital només aquell usuari amb aquell certificat podrà executar transaccions amb aquell participant.
- GET */participant/:id*: S'obté la identitat d'un participant, sense donar dades sensibles. Això serveix per poder oferir al client final informació sobre els intermediaris que hi ha hagut al llarg de la cadena de subministrament.
- POST */crop*: Mètode que permet la creació d'un objecte camp de cafè. Se li ha de passar com a paràmetres la id de l'objecte a crear, el propietari, la data de creació, el nom de la granja, la geoposició i el país. Com en el cas del participant, s'associarà la identitat del creador, de manera que es rebutgi qualsevol canvi a l'objecte a no ser que sigui el seu titular.
- POST */sack*, */bag*, */cup*: Mètode que permet la creació d'un objecte. Se li passa com a paràmetre la id del pare (del crop si és un sack, del sack si és un bag o del bag si és un cup), la id de creació de l'objecte, la id del seu propietari i la data de creació. El sistema verificarà que la id de participant que se li passa està associada al certificat digital de qui fa la transacció i que efectivament sigui propietari del camp de cafè. Pel cas de la bossa de cafè o de la tassa de cafè és totalment anàleg.
- GET */crop/:id*, */sack/:id*, */bag/:id*, */cup/:id*: Obté l'objecte amb la id que se li passa com a paràmetre. Serveix per obtenir els paràmetres de qualitat associats al cafè.
- POST */crop/parameters*, */sack/parameters*, */bag/parameters*, */cup/parameters*: Mètode que rep per post la ID crop de l'objecte i una estructura de dades on hi ha un o diversos paràmetres de qualitat del cafè. Cal dir que hi ha paràmetres del cafè que són *readonly*, és a dir, que un cop inserits no es permet noves modificacions. D'altres en canvi, com la temperatura, són arrays on cada cop s'insereixen diversos valors associats a un *timestamp*. Pel cas del sac i la bossa de cafè és totalment anàleg. L'objecte cup no té opció a posar-hi paràmetres ja que és l'objecte final resultat de tots els processos anteriors.

- POST */crop/set-harvested*: Mètode per indicar quants quilos de cafè s'han cultivat en el camp.
- POST */sack/update-geo*, */bag/update-geo*: Mètode que permet actualitzar la geolocalització dels sacs i les bosses de cafè. No aplica al camp, ja que no es mourà de lloc un cop creat com tampoc a les tasses de cafè, que un cop es creen es consumeixen al mateix espai.
- POST */sack/transfer*, */bag/transfer*: Mètode que permet transferir un objecte des d'un usuari a un altre. Se li passarà per post la ID de l'usuari al qual se li vol transferir i la ID de l'objecte. Referent a l'ABAC, només l'usuari propietari d'aquell bé el podrà transferir.
- GET */cup/farmer/:id*: Aquest mètode el que fa és retornar, donada una tassa de cafè, totes les tasses de cafè produïdes amb el cafè d'un mateix agricultor. La seva utilitat és poder oferir més informació al client final en el moment del consum de la tassa.
- POST */crop/close*, */sack/close*, */bag/close*, */cup/close*: Mètode que permet tancar un objecte per tal de que no es pugui modificar més. Generalment passa quan s'acaben tots els quilos de cafè que hi havia en stock per crear objectes fills.

# Capítol 8

## Implementació del prototip

Si en l'apartat anterior s'ha dissenyat la xarxa Hyperledger, el *chaincode* i la API en aquest s'explicarà l'aplicació del disseny a un prototip per validar el seu funcionament.

Cal dir que la implementació que es fa en aquest apartat no és exactament el que s'ha descrit en l'apartat de disseny, ja que l'objectiu del TFG, com ja es va establir en l'apartat d'abast (2), és el d'implementar el codi del *chaincode* i demostrar el seu correcte funcionament en un entorn tancat local, a l'hora que es configura la xarxa i es genera una API per interaccionar-hi. Es farà sobre una distribució Linux Ubuntu versió 18.04, amb Node.js versió 8.15.1 i docker 18.09.6.

Els principals canvis respecte del disseny és que al no portar-se a producció la xarxa Blockchain que s'implementa és en un sol node (el propi que executa el codi) i l'Ordering Service està implementat com a *Solo*. A més, l'índex de qualitat del cafè s'ha tengut en compte per modelar la informació que cal recollir en la Blockchain, però al ser un prototip no s'han inclòs tots els paràmetres, pel que no es pot donar una qualificació final. Amb tot, s'aniran veient totes les diferències a mesura que s'expliquin les implementacions.

Pel que fa al codi del *chaincode* és exactament el mateix.

### 8.1 Xarxa Hyperledger Fabric

Per implementar una xarxa Hyperledger de desenvolupament el projecte se serveix de Hurley, una eina de Worldsibu per desplegar una Blockchain on testear contractes intel·ligents [10]. A més, Hurley pot funcionar conjuntament amb Convector. S'usarà Hurley 1.0. Els binaris d'Hyperledger utilitzats seran els de la versió 1.4 (l'última en funcionament durant el temps que s'ha implementat el TFG).

Anem a detallar la xarxa que es crearà:

- Una xarxa amb 3 organitzacions. Hurley les anomenarà *org1*, *org2* i *org3*. Cada una d'elles tindrà el seu propi MSP.
- Per simplicitat, cada una d'elles tindrà un sol peer a la xarxa.

- Com s'ha dit, l'Ordering Service utilitzarà Solo en el qual només hi ha un sol node en tota la xarxa que faci la seva funció. Aquesta configuració serà només per a l'execució d'aquest entorn local per fer les proves, no pas per a passar-ho a producció.
- Hi haurà un sol canal de comunicació, que representarà el canal del cafè i serà on s'instal·li posteriorment el codi.

Cada un d'aquests serveis s'executarà en un contenidor docker diferent. Per a fer-ho, executarem la línia següent:

```
$ hurl new -o 3 -u 2 -c 1
```

El que hem fet en aquesta línia ha estat iniciar una nova xarxa amb 3 organitzacions, 2 usuaris per organització i un sol canal. Anem a repassar els arxius que s'han creat per així comprovar la xarxa sobre la qual estarem operant.

El primer fitxer que generem és el *configtx.yaml*

```
1 Organizations:
2   - &OrdererOrg
3     Name: OrdererOrg
4     ID: OrdererMSP
5     MSPDir: ./artifacts/crypto-config/ordererOrganizations/hurley.lab/msp
6
7   - &org1
8     Name: org1MSP
9     ID: org1MSP
10    MSPDir:
11      ↪ ./artifacts/crypto-config/peerOrganizations/org1.hurley.lab/msp
12    AnchorPeers:
13      - Host: peer0.org1.hurley.lab
14        Port: 7051
15
16   - &org2
17     Name: org2MSP
18     ID: org2MSP
19     MSPDir:
20       ↪ ./artifacts/crypto-config/peerOrganizations/org2.hurley.lab/msp
21     AnchorPeers:
22       - Host: peer0.org2.hurley.lab
23         Port: 7051
24
25   - &org3
26     Name: org3MSP
```

```
25     ID: org3MSP
26     MSPDir:
27       ↪ ./artifacts/crypto-config/peerOrganizations/org3.hurley.lab/msp
28     AnchorPeers:
29       - Host: peer0.org3.hurley.lab
30         Port: 7051
31
32 Orderer: &OrdererDefaults
33     OrdererType: solo
34
35     Addresses:
36       - orderer.hurley.lab:7050
37
38 Channel: &ChannelDefaults
39     Capabilities:
40       <<: *ChannelCapabilities
41
42 OrgsChannel:
43     Consortium: SampleConsortium
44     <<: *ChannelDefaults
45     Application:
46       <<: *ApplicationDefaults
47     Capabilities:
48       <<: *ApplicationCapabilities
49     Organizations:
50       - *org1
51       - *org2
52       - *org3
```

Aquí es pot observar que hem creat 3 organitzacions, l'orderer, el canal i el consorci (s'han obviat algunes parts de l'arxiu que no eren importants pel que es volia demostrar).

En segon lloc s'ha generat l'arxiu *crypto-config.yaml*:

```
1 OrdererOrgs:
2   - Name: Orderer
3     Domain: hurley.lab
4     Specs:
5       - Hostname: orderer
6
7 PeerOrgs:
8   - Name: org1
9     Domain: org1.hurley.lab
```



```
10     EnableNodeOUs: true
11     Template:
12       Count: 1
13     Users:
14       Count: 2
15   - Name: org2
16     Domain: org2.hurley.lab
17     EnableNodeOUs: true
18     Template:
19       Count: 1
20     Users:
21       Count: 2
22   - Name: org3
23     Domain: org3.hurley.lab
24     EnableNodeOUs: true
25     Template:
26       Count: 1
27     Users:
28       Count: 2
```

Aquest arxiu es genera juntament amb tot el material criptogràfic necessari per al correcte funcionament de qualsevol entitat dins la xarxa. Tot i que es generen molts més arxius per a la configuració de la xarxa Hyperledger, no és l'objectiu del present treball entrar a analitzar-ho. Es poden mirar els logs dels contenidors de hyperledger per comprovar que la xarxa està funcionant i que està preparada per a instanciar-li el *chaincode*.

## 8.2 Chaincode

Com s'ha dit ja anteriorment s'utilitzarà Convector com a *framework* per escriure el codi del *chaincode*, concretament la versió 1.3.

La implementació del model s'ha fet exactament seguint el disseny de l'apartat anterior. A continuació s'ensenya el model de crop (arxiu crop.model.ts). La resta de models no s'ensenyaran ja que l'estructura que segueixen és molt similar a aquesta:

```
1 export class Crop extends ConvectorModel<Crop> {
2   @ReadOnly()
3   @Required()
4   public readonly type = "io.thethings.crop";
5
6   @ReadOnly()
7   @Required()
```

```
8      @Validate(yup.number())
9      public readonly created: number;
10
11     @Required()
12     @Validate(yup.number())
13     public modified: number;
14
15     @ReadOnly()
16     @Required()
17     @Validate(yup.string())
18     public readonly createdBy: string;
19
20     @Required()
21     @Validate(yup.string())
22     public modifiedBy: string;
23
24     @Required()
25     @Validate(yup.string())
26     public holderId: string;
27
28     @ReadOnly()
29     @Required()
30     @Validate(yup.string())
31     public farm: string;
32
33     @ReadOnly()
34     @Required()
35     @Validate(Geo)
36     public geolocation: FlatConvectorModel<Geo>;
37
38     @Required()
39     @Validate(yup.boolean())
40     @Default(() => false)
41     public isClosed: boolean
42
43     @Validate(Parameters)
44     @Default(() => new Parameters())
45     public parameters: FlatConvectorModel<Parameters>;
46
47     @Validate(yup.number())
48     @Default(() => 0)
49     public harvestedKgCoffee: number;
```

```
50
51 @Validate(yup.number())
52 @Default(() => 0)
53 public remainingKgCoffee: number;
54 }
```

El que ens interessa veure aquí, i que de forma anàloga s'extrapola a la resta de models, és l'ús de la llibreria yup, que permet el control dels tipus de variables i d'estructures definides per l'usuari. D'aquesta manera i combinat amb Typescript, s'eviten fer els controls de variables que sí que s'haurien de fer amb Javascript. Yup va amb el decorador Validate.

Podem veure també els decoradors de ReadOnly, Required i Default. El primer permet que un cop s'escriu un valor a una propietat no es permet tornar a fer cap escriptura nova. Required exigeix a l'usuari que s'insereixi valor a la propietat, mentre que Default omple per defecte la propietat en cas que no l'insereixi l'usuari.

Passem ara a veure algunes funcions importants dels controladors de participant i dels objectes del cafè. A continuació es pot observar la funció que registra un nou usuari a la xarxa:

```
1 @Invokable()
2 public async register(
3   @Param(yup.string())
4   id: string,
5   @Param(yup.string())
6   name: string,
7   @Param(yup.string())
8   role: string
9 ) {
10   const existing = await Participant.getOne(id);
11
12   if (!existing || !existing.id) {
13     let participant = new Participant(id);
14     participant.name = name;
15     participant.msp = this.fullIdentity.getMSPID();
16     participant.role = role;
17
18     // Create a new identity
19     participant.identities = [{
20       fingerprint: this.sender,
21       status: true
22     }];
23     await participant.save();
24   } else {
```

```
25     throw new Error("Identity exists already, please call  
    ↪   changeIdentity for updates");  
26   }  
27 }
```

Aquesta funció ens serveix d'excusa per veure com se guarden models a la Blockchain. A més, podem veure a la línia 15 la funció `getMSPID()`. Aquesta és una funció que Convector adopta i prové directament de l'SDK que Hyperledger Fabric ofereix als desenvolupadors per crear el seu propi codi [11]. Després de comprovar que el participant no existeixi encara, afegeix la identitat a l'array i la posa com a activa. Cal dir que part del codi del participant és reutilitzat de projectes de Wordsibu, que els té tots sota llicència MIT.

Cal fer èmfasi també amb la ID de creació. En general quan en una base de dades es crea un objecte o una entrada és el mateix sistema qui li associa una ID. En aquest cas, però, hem de pensar que el nostre cas d'ús és el de la cadena de subministrament. En el nostre cas, quan un agricultor crea un sac de cafè al sistema el crea en base a una ID ja predefinida que eventualment podria ser en forma de QR i aferrada al sac. Altres sistemes ja funcionen així [14]. Això també hauria de ser així per crear bosses de cafè o tasses.

A continuació tenim una implementació d'una funció molt important del sistema i és una de les que implementa el sistema ABAC:

```
1 public static async checkParticipant(  
2   @Param(yup.string())  
3   id: string,  
4   @Param(yup.string())  
5   sender: string,  
6   @Param(yup.string())  
7   role: string = "no_role"  
8 ) {  
9   const participant = await Participant.getOne(id);  
10  
11   if (!participant || !participant.id || !participant.identities) {  
12     throw new Error("Referenced participant ${id} does not exist in the  
    ↪   ledger");  
13   }  
14  
15   const currentIdentity = participant.activeIdentity();  
16   if (currentIdentity.fingerprint !== sender) {  
17     throw new Error("Participant does not match identity making the  
    ↪   transaction");  
18   }  
19  
20   if(participant.role !== role && role !== "no_role"){
```

```
21     throw new Error("Participant is not a ${role}");
22   }
23
24   return participant;
25 }
```

El que fa aquesta funció és rebre com a paràmetre la ID de l'usuari, enviada pel propi usuari, reb també el fingerprint de qui fa la transacció (que l'usuari no pot controlar, ja que està associat a la identitat digital) i, opcionalment, el rol. Fa el match entre ID d'usuari i identitat i retorna error en cas que no ho faci. D'aquesta manera un usuari no podrà modificar els actius que no siguin de la seva propietat. Finalment, si se li passa com a paràmetre el rol del participant valida també que el tengui. Això ens serveix per evitar que un transportista pugui crear, per exemple, un sac de cafè si no li pertoca.

Revisem ara la funció que transfereix un actiu, ja sigui sac o bossa de cafè:

```
1  @Invokable()
2  public async transfer(
3    @Param(yup.string())
4    bagId: string,
5    @Param(yup.string())
6    to: string
7  ){
8    // Check if the current bag exists
9    let bag = await Bag.getOne(bagId);
10   if (bag.id !== bagId) throw new Error("There is not any bag with this
    ↳ id");
11
12   // Check if the user is the owner
13   let participant = await
    ↳ ParticipantController.checkParticipant(bag.holderId,
    ↳ this.sender);
14   if (bag.isClosed) throw new Error("This object is closed. You can not
    ↳ transfer a closed bag");
15
16   // Check that the id of the receiver exists
17   if ((await Participant.getOne(to)).id !== to) {
18     throw new Error("Tried to assign a bag to a participant that does
    ↳ not exist");
19   }
20
21   let newTransfer = new Transfer({from: bag.holderId, to, datetime:
    ↳ Date.now()});
```

```
22     if (bag.transfers) bag.transfers.push(newTransfer);
23     else bag.transfers = [newTransfer]
24     bag.holderId = to;
25     bag.modifiedBy = this.sender;
26     bag.modified = Date.now();
27     await bag.save();
28 }
```

Com podem observar, el codi comprova que la identitat digital de qui envia la transacció correspongui amb l'usuari que és titular de la bossa de cafè. Comprova també que l'objecte no estigui tancat. Finalment, crea un objecte Transfer nou, l'afegeix i acaba fent la transferència.

Una altra de les funcions més utilitzades en el sistema és precisament la que s'encarrega d'anar actualitzant els paràmetres de qualitat del cafè:

```
1  @Invokable()
2  public async setParameters(
3      @Param(yup.string())
4      sackId: string,
5      @Param(Parameters.schema())
6      params: Parameters
7  ){
8      // Check if the current sack exists
9      let sack = await Sack.getOne(sackId);
10     if (sack.id !== sackId) throw new Error("There is not any sack with
        ↳ this id");
11
12     // Check if the user is the owner
13     let participant = await
        ↳ ParticipantController.checkParticipant(sack.holderId,
        ↳ this.sender);
14     if (sack.isClosed) throw new Error("This object is closed. You can
        ↳ not modify the sack");
15
16     Object.keys(params).forEach((key) => {
17         if(typeof params[key] !== "undefined" && key !== "type"){
18             if(Array.isArray(params[key])){
19                 for(let i=0; i<params[key].length; i++){
20                     ↳ sack.parameters[key].push(params[key][i]);
21                 }
22             }
23             else {
24                 if(typeof sack.parameters[key] !== "undefined") throw new
25                     ↳ Error("You can not redefine the value of ${key}");
26             }
27         }
28     });
29 }
```

```
23     sack.parameters[key] = params[key];
24   }
25 }
26 });
27
28 sack.modified = Date.now();
29 sack.modifiedBy = this.sender;
30 await sack.save();
31 }
```

El que ens interessa veure d'aquí és que la llibreria Yup facilita el fet d'enviar els paràmetres. El backend tindrà la definició de la classe Parameter, de manera que en una sola estructura l'usuari o el dispositiu IoT pot ficar un o diversos paràmetres de qualitat del cafè. En cas d'introduir una propietat que no existeixi el sistema retornarà error.

Finalment es vol mostrar també un altre exemple de funció a fi de mostrar una altra de les implementacions importants del sistema:

```
1  @Invokable()
2  public async getByFarmer(
3    @Param(yup.string())
4    idFarmer: string
5  ) {
6    return await Cup.query(Cup, {
7      "selector": {
8        "type": "io.thethings.cup",
9        "parameters.farmer": idFarmer
10     }
11   });
12 }
```

Com s'ha dit en l'apartat d'anàlisi d'Hyperledger (6) la base de dades que sosté el *world state* és CouchDB, que només té l'últim estat de la Blockchain. Atacar directament la Blockchain és molt més costós que no el CouchDB. Evidentment, la garantia d'obtenir les dades fiables és atacar la Blockchain, però per algunes consultes com un simple GET es pot fer ús de CouchDB. A més, fer-ho així ens permet fer consultes molt més elaborades i a un cost temporal menor. Per atacar CouchDB es pot fer a través de Mango Queries [5].

El codi anterior, per exemple, ens retorna totes les tasses de cafè elaborades amb el cafè d'un mateix agricultor d'id *idFarmer*.

## 8.3 Verificació del chaincode

Per a verificar el *chaincode* s'ha usat la llibreria mocha, que és una llibreria de nodejs per verificar el codi [13]. El que s'ha fet ha estat usar aquesta llibreria per a cridar totes les funcions i emular els resultats sense necessitat de crear cap instància de Blockchain. És útil per testejar el codi de forma aïllada.

Tot i així, això no seria possible si Hyperledger no proveís una llibreria per emular precisament el funcionament d'una xarxa Hyperledger. Aquesta llibreria s'anomena Mock Stub [4] i forma part del core d'Hyperledger i de l'SDK.

Per exemple, veiem el cas del test de l'objecte cup. La gràcia de fer tests unitaris és que es pugui verificar el codi sense ingerències externes. I aquí entra Mock Stub. Ja que volem testejar el controlador de cup, no ens interessa cridar el controlador de participant (és necessari per interaccionar amb qualsevol objecte). Per això introduïrem informació de forma emulada:

```
1 let participantId = "carles@coffeeshop.io";
2 let certificate =
  ↪ "B6:0B:37:7C:DF:D2:7A:08:0B:98:BF:52:A4:2C:DC:4E:CC:70:91:E1";
3 await adapter.stub.mockTransactionStart("mockTx0");
4 await adapter.stub.putState(participantId, Transform.serialize({
5   id: participantId,
6   type: "io.thethings.participant",
7   role: "barista",
8   identities: [{fingerprint: certificate, status: true}]
9 }));
```

El que fem és emular el registre d'un usuari de rol "barista". Primer iniciem una transacció i posteriorment modifiquem l'estat de la Blockchain. Ara que tenim informació a la Blockchain podem testejar funcions reals del controlador de cup.

```
1 it('should create a bag', async () => {
2   await cupCtrl.create(cupId, bagId, participantId, Date.now(), 0.002);
3   const newCup = await adapter.getById<Cup>(cupId);
4   expect(newCup.id).to.eq(cupId);
5   expect(newCup.parameters.terrainAltitude).to.eq(1850);
6 });
7
8 it('should get the cup created', async () => {
9   const cup = await cupCtrl.get(cupId);
10  expect(cup).to.exist;
11 });
12
```



```

13 it('should get coffee cups from the same farmer', async () => {
14   let cups = <Cup[]>(await
    ↪   cupCtrl.getByFarmer('antonia@farmcolombia.io'));
15   expect(cups.length).to.be.eq(1);
16 });

```

El codi que s’ha ensenyat són dues funcions que ja pertanyen a mocha (la funció *it*). En la primera s’hauria de crear una tassa de cafè. Es veu com se li passa per paràmetre la id de la tassa, la id de la bossa de la qual deriva, la id del participant, la data i la quantitat de grams d’una tassa. Així com hem fet introduint informació prèvia amb el participant, també s’ha hagut de fer amb l’objecte bossa de cafè amb id bagId. Veiem com la funció *expect()* valida el resultat que hauria de tornar. En cas que sigui l’esperat el test és superat, en cas que no el test falla i caldria revisar el codi.

En segon lloc comprovem una altra funció que retorna una tassa de cafè donada la seva id. Finalment, la funció feta amb mango queries on es retornen totes les tasses de cafè elaborades amb el cafè de l’agricultor amb id “antonia@farmcolombia.io”. En aquest cas prèviament només s’havia inserit al sistema una tassa de cafè, pel que s’espera que el resultat sigui 1.

La sortida en aquest cas, després d’executar el test de cup, és la següent:

```

cup-cc:      ✓ should create a cup (81ms)
cup-cc: info: [Chaincode] ===== Invoked Chaincode Chaincode =====
cup-cc: info: [Chaincode] Transaction ID: 9b3b49d0-86d9-11e9-ad8e-a15a17580632
cup-cc: info: [Chaincode] Args: cup_get,5a0dc71a-e0ca-4f96-bdcf-26fcb36982c0
cup-cc: debug: [Chaincode] ===== START : cup_get =====
cup-cc: debug: [Chaincode] ===== END : cup_get =====
cup-cc: debug: [Chaincode] ===== START : cup_getByFarmer =====
cup-cc: debug: [StubHelper] Query: {"selector":{"type":"io.thethings.cup"},"parameters.farmer":"antonia@farmcolombia.io"}}
cup-cc:      ✓ should get the cup created
cup-cc: info: [Chaincode] ===== Invoked Chaincode Chaincode =====
cup-cc: info: [Chaincode] Transaction ID: 9b3c8250-86d9-11e9-ad8e-a15a17580632
cup-cc: info: [Chaincode] Args: cup_getByFarmer,antonia@farmcolombia.io
cup-cc: debug: [Chaincode] ===== END : cup_getByFarmer =====
cup-cc:      ✓ should get coffee cups from the same farmer
cup-cc:    3 passing (291ms)
lerna success run Ran npm script 'test' in 1 package in 10.7s:
lerna success - cup-cc

```

Figura 8.1: Test mocha pel controlador de *cup*

## 8.4 Implementació de l’API

En el projecte també s’ha implementat una primera versió de l’API, pensada per funcionar en l’entorn local. Per la versió del prototip s’han implementat les funcions via API del

participant, i dels models de crop, sack i cup. S'ha fet amb el paquet Express de nodejs, que s'utilitza per crear aplicacions web i API [3].

L'arquitectura és la següent (font WorldSibu):

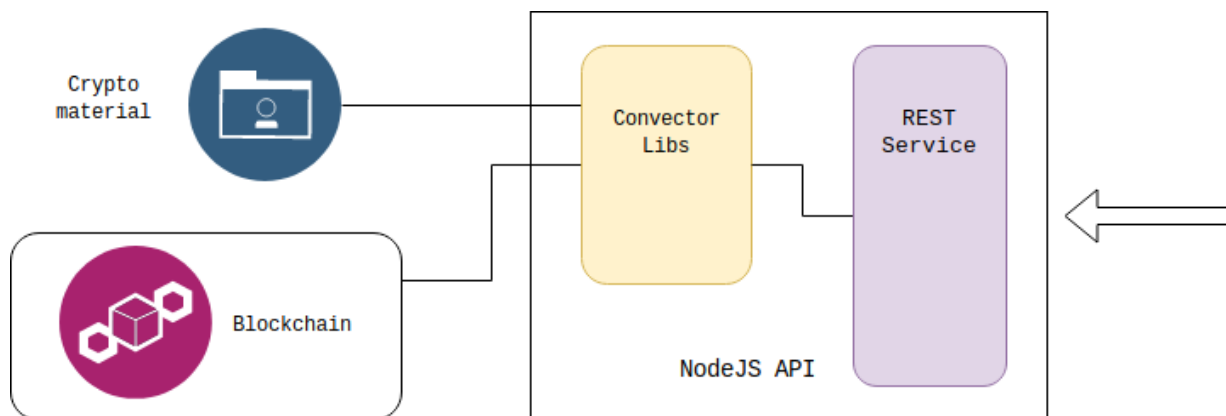


Figura 8.2: Arquitectura de l'aplicació

Més enllà dels arxius de configuració, que podrien ser iguals en tants d'altres projectes, ens interessa aquí veure la implementació de l'arxiu dels controladors. En les següents línies de l'arxiu `app.ts` establim la ruta de cada controlador:

```
1 app.use('/participant', ParticipantExpressController);
2 app.use('/cup', CupExpressController);
3 app.use('/crop', CropExpressController);
```

En aquestes línies es pot veure com l'aplicació utilitza un controlador per a cada tipus de url. Anem, doncs, a veure algunes de les funcions d'aquests controladors:

```
1 router.get('/:id', async (req: Request, res: Response) => {
2   try {
3     let { id } = req.params;
4     res.send(await ParticipantControllerBackend.get(id));
5   } catch (err) {
6     console.log(JSON.stringify(err));
7     res.status(500).send(err);
8   }
9 });
```

Aquesta és una funció molt simple i que habilita a “/participant/:id” un mètode de tipus GET per obtenir el participant donada una id.

```
1 router.post('/', async (req: Request, res: Response) => {
2   try {
3     const { cropId, owner, created, farm, lat, lon, country } =
4       ↪ req.body;
5     await CropControllerBackEnd.create(cropId, owner, created, farm,
6       ↪ lat, lon, country);
7     res.status(201).send({'Status': 200, 'Response': 'Crop
8       ↪ created'});
9   } catch (err) {
10     console.log(JSON.stringify(err));
11     res.status(500).send(err);
12   }
13 });
14
15 router.post('/parameters', async (req: Request, res: Response) => {
16   try {
17     const {cropId, parameters} = req.body;
18     const newParameters = new Parameters(parameters);
19     await CropControllerBackEnd.setParameters(cropId, newParameters);
20     res.status(201).send({'Status': 200, 'Response': 'Parameters
21       ↪ updated'});
22   } catch (err) {
23     console.log(JSON.stringify(err));
24     res.status(500).send(err);
25   }
26 });
```

Les dues últimes funcions són dos POST, el primer per crear un objecte de tipus crop i el segon per introduir-hi paràmetres. Veiem com l'API també utilitza els models creats en el propi *chaincode*, de manera que facilita molt la feina a l'hora d'unir una capa amb l'altra.

## Capítol 9

# Verificació integral

Per tal de fer la verificació integral del projecte s'han afegit utilitats per tal de veure que tot funciona correctament. Primer de tot instal·larem el chaincode:

```
$ npm run cc:start -- coffee
```

Aquesta línia instal·larà el *chaincode*. En cas que no haguéssim creat encara la xarxa Blockchain, seria el moment de fer-ho (s'ha fet en l'apartat 8). Sinó, es podria fer igualment amb la línia següent:

```
$ npm run env:restart
```

Ara que tenim el codi instal·lat podem interaccionar amb ell. S'ha preparat un script per iniciar la Blockchain amb dades d'exemple. L'script és el següent:

```
echo "Seeding the blockchain with some data..."
hurl invoke coffee participant_register "carles@farm.io" "Carles"
↳ "farmer"

hurl invoke coffee crop_create "crop1" "carles@farm.io" 1554209553 "Farm
↳ Colombia" 3.304296295969481 -- -76.49315399115159 "Colombia"

hurl invoke coffee crop_setParameters "crop1" "{\"terrainAltitude\":
↳ 1900, \"landCharacteristics\": \"very well oxygenated\", \"variety\":
↳ \"arabica\", \"classificationCherriesDay\": 1554239270,
↳ \"fermentationGrainDay\": 1554239999 }"

hurl invoke coffee crop_setHarvestedKgCoffee "crop1" 200

hurl invoke coffee sack_create "sack1" "crop1" "carles@farm.io"
↳ 1555332753 20
```

```

hurl invoke coffee sack_setParameters "sack1" "{\"storageGrainDay\":
↳ 1555419153}"

hurl invoke coffee sack_updateGeo "sack1" "[{\"lat\": 5.6225230562200785,
↳ \"lon\": -79.88792938177659, \"datetime\": 1555505553}, {\"lat\":
↳ 11.267463, \"lon\": -77.152334, \"datetime\": 1555678353}, {\"lat\":
↳ 51.952425, \"lon\": 4.088603, \"datetime\": 1556283153}]"

hurl invoke coffee participant_register "antonia@transport.io" "Antonia"
↳ "transportist" -u admin

hurl invoke coffee sack_transfer "sack1" "antonia@transport.io"

hurl invoke coffee sack_transfer "sack1" "carles@farm.io" -u admin

hurl invoke coffee bag_create "bag1" "sack1" "carles@farm.io" 1556369553
↳ 5

hurl invoke coffee bag_updateGeo "bag1" "[{\"lat\": 41.411403, \"lon\":
↳ 2.177948, \"datetime\": 1555505553}]"

hurl invoke coffee cup_create "cup1" "bag1" "carles@farm.io" 1556801553
↳ 20

```

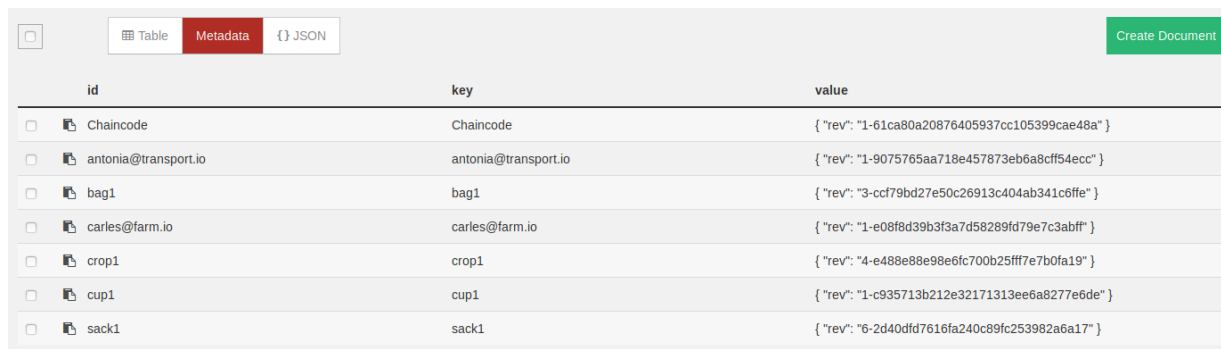
El que s’ha fet en aquest script és usar Hurl (amb el que ja vam iniciar la xarxa amb els paràmetres descrits en l’apartat anterior) per inserir dades directament. La comanda *invoke* serveix per cridar una funció d’un controlador. El paràmetre *coffee* fa referència al nom del *chaincode* que hem instal·lat abans. Finalment es posa la funció com a “nomControlador\_nomFunció”.

D’aquesta manera podem veure com primer es crea un participant amb la ID “carles@farm.io”, de nom Carles i amb el rol granger. Posteriorment aquest granger crea un objecte crop amb els paràmetres que necessita. També se li assignen una sèrie de paràmetres com l’altitud del terreny, característiques de la terra, la veritat del cafè o dates de classificació i fermentació de les cireres. Per acabar, aquest granger introdueix que ha recollit 200kg de cafè i crea un sac de 20kg.

Aquest sac és també actualitzat amb una nova geolocalització i més paràmetres. En aquest moment es registra un nou participant. Cal veure aquí que qui crea aquest nou participant és l’usuari administrador de la xarxa blockchain. Si no s’estipula, hurl fa les crides amb l’usuari *user1* de l’organització *org1*. Ara el granger, titular d’aquell sac, el transfereix a un transportista.

Finalment es crea també una bossa de 5kg i una tassa de cafè de 20 grams. Tot, amb el cafè del mateix granger.

Com es pot verificar que tot és correcte? Hurl aixeca l'adreça `http://localhost:5084/_utils/#database/ch1_coffee/_all_docs` on es pot consultar el *world state*. En la següent imatge podem veure com s'han creat tots els objectes que s'han especificat a l'script:



id	key	value
<input type="checkbox"/> Chaincode	Chaincode	{ "rev": "1-61ca80a20876405937cc105399cae48a" }
<input type="checkbox"/> antonia@transport.io	antonia@transport.io	{ "rev": "1-9075765aa718e457873eb6a8cff54ecc" }
<input type="checkbox"/> bag1	bag1	{ "rev": "3-ccf79bd27e50c26913c404ab341c6ffe" }
<input type="checkbox"/> carles@farm.io	carles@farm.io	{ "rev": "1-e08f8d39b3f3a7d58289fd79e7c3abff" }
<input type="checkbox"/> crop1	crop1	{ "rev": "4-e488e88e98e6fc700b25fff7e7b0fa19" }
<input type="checkbox"/> cup1	cup1	{ "rev": "1-c935713b212e32171313ee6a8277e6de" }
<input type="checkbox"/> sack1	sack1	{ "rev": "6-2d40dfd7616fa240c89fc253982a6a17" }

Figura 9.1: World State de la Blockchain

Si cliquem a l'objecte tassa de cafè veurem que ha heretat tots els paràmetres dels objectes pare, així com l'històric de geolocalitzacions i transferències:

```
{
  "_id": "cup1",
  "_rev": "1-c935713b212e32171313ee6a8277e6de",
  "bagId": "bag1",
  "created": 1556801553,
  "createdBy":
    ↪ "FA:01:12:44:19:01:4B:23:FB:8F:89:A8:42:54:6A:3A:F0:8C:95:A0",
  "grams": 20,
  "holderId": "carles@farm.io",
  "modified": 1556801553,
  "modifiedBy":
    ↪ "FA:01:12:44:19:01:4B:23:FB:8F:89:A8:42:54:6A:3A:F0:8C:95:A0",
  "parameters": {
    "classificationCherriesDay": 1554239270,
    "farm": "Farm Colombia",
    "farmCountry": "Colombia",
    "farmer": "carles@farm.io",
    "fermentationGrainDay": 1554239999,
    "landCharacteristics": "very well oxygenated",
    "route": [
      {"datetime": 1554209553, "lat": 3.304296295969481, "lon":
        ↪ -76.493153},
      {"datetime": 1555505553, "lat": 5.6225230562200785, "lon":
        ↪ -79.88799},
    ]
  }
}
```

```

    {"datetime": 1555678353, "lat": 11.267463, "lon": -77.152334},
    {"datetime": 1556283153, "lat": 51.952425, "lon": 4.088603},
    {"datetime": 1555505553, "lat": 41.411403, "lon": 2.177948}
  ],
  "storageGrainDay": 1555419153,
  "terrainAltitude": 1900,
  "thermalAmplitude": [],
  "type": "io.thethings.parameters",
  "variety": "arabica"
},
"transfers": [
  {"datetime": 1559744553693, "from": "carles@farm.io", "to":
    ↪ "antonia@transport.io", "type": "io.thethings.transfer"},
  {"datetime": 1559744558686, "from": "antonia@transport.io", "to":
    ↪ "carles@farm.io", "type": "io.thethings.transfer"}
],
"type": "io.thethings.cup",
"~version": "\u0000CgMBEAA="
}

```

Fins aquí hem vist que tot funciona correctament. S'ha afegit al projecte la utilitat KHS Blockchain Browser, que serveix per anar veient en temps real la creació dels blocs en una xarxa blockchain. S'ha utilitzat el fork que ha fet Worldsibu [8]. A continuació podem veure la creació del bloc gènesi (explicat a l'apartat 6):

Block	Blockchain Info	Peers
<a href="#">Raw</a>   <a href="#">Genesis</a>   <a href="#">Latest</a> <b>Number:</b> 1/18 ➔ <b>Type:</b> CONFIG <b>Block Hash:</b> fae5c91e154f7157d17b794671245d0fb5d37d58c0bb52dbcf9f705c01420ad3 <b>Previous Hash:</b> <b>Timestamp:</b> Wed Jun 05 2019 16:01:02 GMT+0200 (CEST) <b>Transactions :</b> 1	<b>Channel:</b> ch1 <b># Blocks:</b> 18 <b>Chaincodes:</b> name: coffee version: 1.0	• Peer: {url:grpc://localhost:7051}
<b>Configuration</b>		
<b>Orderer</b> <b>Consensus Type:</b> solo <b>Batch Size:</b> 10 <b>Batch Timeout:</b> 2s	<b>Channel</b> <b>Hashing Algorithm:</b> SHA256 <b>Batch Size:</b> 10	

Figura 9.2: Creació del bloc gènesi

O, en aquest altre, podem veure com sí que s'ha creat el bloc amb la transacció corresponent a la creació de bag1:

### Block

[Raw](#) | [Genesis](#) | [Latest](#)

**Number:** ◀ 17/18 ▶

**Type:** ENDORSER\_TRANSACTION

**Block Hash:**  
35f4f70f694b251563b69819da11b08490582d94a88cc08566f03e8c1dad86b7

**Previous Hash:**  
882bc564190fcb718d5a7224c9870fe9f3da1778102c5b1a8bad02d103f431f3

**Timestamp:** Wed Jun 05 2019 16:22:53 GMT+0200 (CEST)

**Transactions :** 1

### Blockchain Info

**Channel:** ch1

**# Blocks:** 18

**Chaincodes:**  
name: coffee version: 1.0

### Peers

• Peer: {url:grpc://localhost:7051}

### Transactions

	Tx Id	Type	Created By	Chaincode	Endorsements
▼	13c022e7053424ea4b816...	ENDORSER_TRANSACTION	org1MSP	coffee(GOLANG)	org1MSPorg2MSP
<b>Reads</b>					
Namespace: coffee					
bag1 = undefined					
Namespace: lsc					
coffee = undefined					
<b>Writes</b>					
Namespace: coffee					
bag1 =					
{ "id": "bag1", "type": "io.thethings.bag", "sackId": "sack1", "created": 1556369553, "modified": 1559744574039, "createdBy": "FA:01:12:44:19:01:4B:23:FB:8F:89:A8:42:54:6A:3A:F0:8C:95:A0",					
[ { "datetime": 1559744553693, "from": "carles@farm.io", "to": "antonia@transport.io", "type": "io.thethings.transfer",					
[ { "datetime": 1559744558686, "from": "antonia@transport.io", "to": "carles@farm.io", "type": "io.thethings.transfer" }, ], "parameters":					
{ "classificationCherriesDay": 1554239270, "farm": "Farm					
Colombia", "farmCountry": "Colombia", "farmer": "carles@farm.io", "fermentationGrainDay": 1554239999, "landCharacteristics": "very well oxygenated", "route":					

Figura 9.3: Creació del bloc de *bag1*

I finalment, verificarem que el servei de la API funciona amb una crida amb curl i l'obtenció del resultat:

```
# Start the server
$ npx lerna run start --scope server --stream
# Get the user carles@farm.io
$ curl http://localhost:8000/participant/carles@farm.io
{"_id": "carles@farm.io", "_identities": [{"fingerprint":
  ↪ "FA:01:12:44:19:01:4B:23:FB:8F:89:A8:42:54:6A:3A:F0:8C:95:A0",
  ↪ "status": true}], "_msp": "org1MSP", "_name": "Carles", "_role":
  ↪ "farmer", "_type": "io.thethings.participant"}
```

O una altra crida que també hem implementat. Inserir una nova tassa de cafè amb el cafè de *bag1*:

```
# Insert a new cup
$ curl -H "Content-Type: application/json" --request POST --data '{
  ↪ "cupId": "cup2", "bagId": "bag1", "holderId": "carles@farm.io",
  ↪ "created": 1556801553, "grams": 20}' http://localhost:8000/cup
```



```
{"Status": 200, "Response": "Cup created"}
```

D'aquesta manera, per tant, es verifica que el codi funciona i que l'API està responnent i que està correctament connectada amb la xarxa Blockchain.

## 9.1 Connexió amb thethings.iO

Tot i que inicialment no estava previst, s'ha afegit codi a allò dissenyat i plantejat inicialment per tal de fer la connexió amb la plataforma thethings.iO. Molt probablement la implementació final es modifiqui substancialment, però afegint aquest codi a la API es poden visualitzar les dades gràficament tal i com les visualitzaria el client final a través d'un ordinador portàtil o de sobretaula. En cas d'aplicació mòbil la visualització seria molt semblant.

Per a fer-ho es necessiten fer dues tasques noves. Per una banda cal modificar la API per tal de que quan la Blockchain retorna la confirmació de que les dades s'han actualitzat correctament les envii a thethings.iO. Finalment, cal afegir a la plataforma el codi encarregat de traduir les dades que es reben per a poder ser visualitzades.

En primer lloc s'afegeix a l'API de la Blockchain la crida següent a l'API de thethings.iO:

```

1 export function SendDataToTTIO(payload){
2   return new Promise((resolve, reject) => {
3     const options = {
4       url: 'https://subscription.thethings.io/http/' +
          ↳ productIdTTIO + '/' + hashTTIO +
          ↳ '/?idname=dev_id&fname=http_parser_TFG&dev_id=' +
          ↳ payload.id,
5       json: true,
6       body: payload
7     };
8
9     request.post(options, (err, res, body) => {
10       if(err) reject(err);
11       else resolve(body);
12     });
13   });
14 }
```

Aquí es pot apreciar que s'usen “productIdTTIO”, que és la identificació interna a thethings.iO del producte creat a la plataforma (en aquest cas, un producte nou en el perfil del client del cafè). També hi ha el hash associat. Aquestes dues són constants

privades a les quals només hi té accés el client. Es pot veure també que es passen com a paràmetres el “devId” i “fname”. El primer és la ID que identifica el dispositiu a la plataforma. Per defecte en aquest cas agafa la ID que té assignada a la Blockchain. El primer cop que envii un missatge a la plataforma es crearà amb aquesta ID. El segon paràmetre fa referència al nom de la funció encarregada de traduir les dades a plataforma.

Com a norma general, els missatges rebuts a plataforma són en hexadecimal, ja que els dispositius IoT actuen a baix nivell. Però també es contempla la possibilitat d’enviar missatges per POST directament en json. Aquest és el cas que ens ocupa, ja que el missatge l’envia l’API, no directament el dispositiu. Així doncs, part de la funció que fa de traductor és la següent:

```
1 function main(params, callback){
2   let payload = params.payload;
3   let result = [];
4
5   // Ruta
6   if(payload.geolocations){
7     for(let i=0; i<payload.geolocations.length; i++){
8       result.push(
9         {
10          "key": "geolocation",
11          "value": "gps",
12          "geo": {"lat": payload.geolocations[i].lat, "long":
13                ↪ payload.geolocations[i].lon},
14          "datetime": moment(payload.geolocations[i].datetime *
15                ↪ 1000).toJSON()
16        },
17        {
18          "key": "$geo", "value": [payload.geolocations[i].lon,
19                ↪ payload.geolocations[i].lat]
20        }
21      );
22    }
23  }
24
25  callback(null, result);
26 }
```

En aquest cas es comprova que el missatge rebut porti informació referent a la ruta i la retorna en una estructura de dades que la plataforma entén per tal de representar la informació gràficament.

La visualització a la plataforma es pot veure en la figura 9.4. Aquí es pot apreciar la

ruta seguida des d'una suposada plantació de cafè a Costa Rica fins al port de Rotterdam. A la figura 9.5 es pot veure, per exemple, la gràfica de temperatura que es dibuixaria per a un dispositiu. Les dades que es mostren aquí són les dades rebudes per l'script que s'ha mostrat en apartats anteriors.

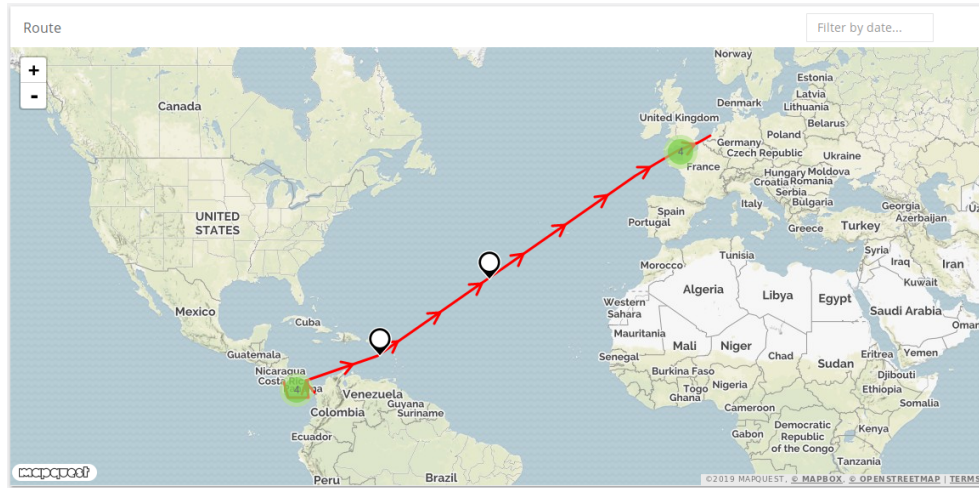


Figura 9.4: Ruta d'un sac de cafè



Figura 9.5: Temperatura d'un sac de cafè

# Capítol 10

## Sostenibilitat i compromís social

En aquest apartat s'estudiarà i aplicarà la matriu de sostenibilitat per valorar i reflexionar sobre les dimensions econòmica, ambiental i social del projecte.

### 10.1 Dimensió econòmica

A continuació es farà una reflexió sobre l'apartat de gestió econòmica. Es considera que el pressupost és assumible, ja que pràcticament no usa més recursos que els humans dels que disposa l'empresa, els de hardware, ja que són els necessaris per a qualsevol empresa tecnològica i els indirectes. L'execució del TFG no implica un gran volum de despeses extra, pel que el fa molt apte com a projecte d'I+D+i de l'empresa i com a prova pilot pel nou servei que haurien de treure al mercat.

Es pot afirmar, a més, que la posada en producció de la solució que es proposa en aquestes pàgines acaba implicant una reducció de pressupostos de l'empresa ja que s'integra en una mateixa plataforma tot el seguiment de la cadena de subministrament. Al no tenir competència en aquests mateixos instants no es pot fer una comparativa econòmica sobre aquesta solució i les altres.

Es considera també que el pressupost ja de per si és ajustat. Treballant-hi pràcticament només una sola persona, pel que seria complicat desplegar un projecte d'aquestes característiques en menys temps i menys pressupost.

És un projecte que, com s'ha dit, es farà de la mà amb Cafès Barcelona i Fair Trade Coffee, malgrat que la implementació del codi i la solució tecnològica vagi estrictament a càrrec de thethings.io.

Finalment, cal dir que quan es porti a producció sí que es podrien produir escenaris en els quals es posés en perill la viabilitat del projecte. L'aportació de capital en la hipotètica posada en producció no serà només responsabilitat de thethings.io, pel que si una de les tres empreses mencionades falla, el projecte podria no portar-se a producció.

## 10.2 Dimensió ambiental

Pel que fa a la dimensió ambiental, cal dir que l'empresa no ha estimat l'impacte que tindria el projecte en producció. Malgrat que no s'hagi fet aquest estudi amb els indicadors que pertocuen sí que es pot veure que els recursos necessaris per portar-los a terme seran els servidors que despleguin la xarxa Blockchain, el consum dels telèfons mòbils que interaccionin amb la xarxa i la despesa d'energia extra de tenir sensors i dispositius també enviant dades. Malgrat tot, cal dir que aquestes despeses no tenen un consum elevat.

És més, aquestes despeses energètiques no serien exclusives per la plataforma en producció i els recursos que s'usen per programar tota la plataforma són reutilitzats o bé compartits dins la pròpia empresa. Això significa que no hi ha pràcticament despeses afegides (transport o alguna altra despesa menor) i que tots els recursos utilitzats se seguiran utilitzant en acabar el projecte.

Com s'ha dit, la plataforma integrarà en una de sola tota la gestió de la cadena de subministrament del cafè. Per tant, evitarà l'ús de diverses tecnologies com també l'ús del paper i altres recursos amb un impacte directe sobre el medi ambient.

És més, en casos com la producció del cafè, on tradicionalment el cultiu ha estat extensiu devastant hectàrees de selva o bosc, un projecte així té també un impacte positiu. Es tracta de visibilitzar cafè d'alta qualitat i amb condicions dignes per persones i medi. Que un projecte així tenguí èxit significa premiar petits productors amb parcel·les petites de cultiu que cuiden la terra i la seva regeneració.

## 10.3 Dimensió social

Pel que fa a la dimensió social cal dir que el projecte ha aportat i està aportant a l'estudiant un gran coneixement que no es tenia anteriorment, tant pel que fa a la vessant tecnològica, doncs no s'havia treballat mai amb Blockchain, com també social. Treballar amb aquest cas d'ús, el del cafè, aporta una visió integral i amb perspectiva històrica de la indústria del cafè: com s'ha cultivat tradicionalment, com s'ha explotat, quines han estat les conseqüències pels seus cultivadors. Però també aspectes qualitatius: quines són les característiques del cafè de qualitat, com s'elabora, quines classes de cafè hi ha, entre d'altres.

Un segon element d'aquesta dimensió és com millora socialment les condicions dels actors implicats. Ja se n'ha començat a parlar en l'anterior apartat, però la plataforma té per objectiu millorar les condicions materials de vida dels treballadors involucrats i, en especial, el productor de cafè. I això es vol fer, com ja s'ha anat repetint, portant tota la informació del procés productiu al client final, fent-lo coneixedor de totes les variables que conformen un cafè de qualitat (condicions físiques, però també socials i ambientals). Com afirma Bennett, "un mercat basat en Blockchain permetria als compradors i venedors comparar les seves negociacions amb informació similar sobre altres transaccions. Agricultors d'arreu del món podrien determinar si els surt a compte o no vendre la seva collita a determinat preu" [30].

Finalment, cal dir que efectivament existeix una necessitat real de canviar les condicions

de treball d'aquests productors i de dotar de valor real el cafè de comerç just. Es pot abordar des de múltiples perspectives, però s'entén que el projecte proposa una solució per aportar un gra de sorra més en aquest canvi i que, per tant, sí que és necessari.

No es creu tampoc que el projecte acabi generant una situació de dependència, sobretot per part dels agricultors. En tot cas, el grau de dependència ara és major, ja que no tenen control del procés productiu una vegada venen el seu producte. Com més eines tinguin a l'abast i informació sobre els preus i consum del cafè al llarg de la cadena de subministrament més empoderament tendran els agricultors.

# Capítol 11

## Lleis i regulacions

Atès que l'empresa *thethings.iO* està localitzada a Catalunya es regeix pel marc legal espanyol i europeu. Així doncs, en aquest apartat es farà una anàlisi d'aquelles lleis i regulacions que afecten el projecte que s'ha exposat en aquestes pàgines. Tenir en compte aquestes normatives és imperatiu a fi d'evitar qualsevol tipus de vulneració dels drets dels usuaris de l'aplicació. El que no es pot estudiar en aquest TFG són les normatives i regulacions que afectin els mateixos productors en el país d'origen, ja que en funció de quin sigui –i no serà cap país de la Unió Europea– estarà regulat per les seves pròpies lleis també, que caldrà respectar en cada cas.

En primer lloc, cal ser conscient que, tot i que l'objectiu del TFG és la presentació d'un prototip, la posada en producció del projecte implica que els usuaris introduiran informació personal a la xarxa Blockchain. És així perquè al ser de tipus *permissioned* cal verificar qui és cada usuari. Dit en altres paraules, la creació d'usuaris amb informació personal serà essencial. Així doncs, una primera legislació que afecta el projecte és el Reglament General de Protecció de Dades [28] (GDPR, en les seves sigles en anglès) que afecta els països de la Unió Europea. Es va aprovar el 2016 i va entrar en vigor el 2018.

Els principis que regeixen la GDPR són els de *accountability*, és a dir, les organitzacions hauran de ser capaces de demostrar que compleixen amb les exigències de la llei. En segon lloc, el principi de protecció de dades per defecte i des del disseny. En altres paraules, s'han d'adoptar mesures per complir amb el reglament des del mateix moment que es dissenya un producte, servei o empresa. Finalment, el principi de transparència.

Algunes de les obligacions que se'n deriven són avaluacions d'impacte sobre la privacitat, les fallides de seguretat hauran de ser informades a les autoritats pertinents, s'amplien també les dades considerades sensibles a dades genèriques i dades biomètriques, entre d'altres. S'imposen també sancions per incompliment de fins a un màxim de 20 milions d'euros o el 4% de la facturació anual.

Una segona llei que afecta directament el projecte és la Llei Orgànica 3/2018 de Protecció de Dades de l'Estat espanyol (LOPD) [26]. L'objectiu principal d'aquesta llei era el d'adaptar la GDPR al marc espanyol (tot i que ha servit, a més, per altres temes no inclosos inicialment). Bàsicament, la llei afirma que el dret a la protecció de dades es regeix directament amb el que dictamina la GDPR i les noves disposicions de la LOPD.

A banda de tot això, el projecte utilitzarà dispositius IoT. Actualment a Europa no hi ha una reglamentació tan exhaustiva com hi pugui haver als Estats Units. A Europa, de moment, hi ha un document de treball anomenat *Advancing the Internet of Things in Europe* [27] i que posa les bases de les futures regulacions: un ecosistema IoT pròsper, un enfocament IoT centrat en les persones i un mercat únic per a les IoT. En tot cas i a dia d'avui, això no es tradueix en cap efecte jurídic sobre el projecte.



# Capítol 12

## Revisió dels compromisos inicials

A l'inici del present treball es va estimar el pressupost del projecte així com la temporalitat. Una vegada s'ha acabat la implementació d'allò planificat inicialment es dedicaran unes línies a revisar els compromisos inicials i avaluar si s'han complert o no.

### 12.1 Planificació temporal

L'estimació de les tasques i les hores dedicades a cada una d'elles ha estat, en línies generals, encertada. Les hores dedicades a cada un dels grans apartats (GEP, disseny, implementació, testing i redacció) s'han respectat. En alguna de les subfases s'ha dedicat més temps a alguna d'elles quan alguna altra ha estat més ràpida d'implementar, però sense alterar substancialment el diagrama de Gantt.

Potser una de les desviacions més visuals ha estat el de la redacció de la memòria final. Si bé estava estipulat que es fes entre el 9 de juny i el 23, el cert és que a mesura que s'anava avançant en el treball s'anaven fent anotacions o escrivint apartats aprofitant la frescor del coneixement. D'aquesta manera, tot i que les hores dedicades hauran estat les mateixes, han estat repartides al llarg del darrer mes i mig (amb una concentració òbvia de cara al juny).

### 12.2 Gestió econòmica

El pressupost del projecte sí que no ha sofert variacions respecte allò pressupostat inicialment. El fet de no haver-hi desviacions temporals ha propiciat que no hi hagués tampoc desviacions econòmiques. De fet, no ha estat necessari res d'allò previst de contingència i imprevistos.

# Capítol 13

## Conclusions

L'objectiu d'aquest Treball Final de Grau ha estat el d'estudiar la problemàtica que presentava un cas d'ús concret per a l'empresa *thethings.iO* i presentar una solució que no només solventés aquest cas, sinó que fos genèrica pel model de negoci de l'empresa. S'ha estudiat així quin era el principal problema i de quines formes tècniques es podia solucionar. S'ha optat per Blockchain com a tecnologia i per Hyperledger Fabric com a plataforma.

Al final del TFG s'ha presentat un prototip funcional que combinava Hyperledger Fabric, Convector, Hurley i una API per interaccionar amb el sistema. S'ha validat el codi del *chaincode* i s'ha arribat a la conclusió que el prototip funciona, de manera que estaria argumentada la seva inversió per a portar-ho a producció. A més, tot i que inicialment no estava previst s'ha enllaçat el projecte de Blockchain amb la plataforma *thethings.iO* per tal de visualitzar les dades gràficament.

És per això que es creu que les decisions tècniques que s'han pres han estat encertades i que per tant el desenvolupament a *thethings.iO* d'una línia de serveis basada en el *supply chain* o cadena de subministrament amb Blockchain seria una decisió encertada des del punt de vista tecnològic.

A més, es conclou que els objectius que s'havien proposat en el capítol 2 d'Abast s'han complert tots dins els terminis establerts. És a dir, el disseny del *chaincode* amb el *framework* Convector; disseny de la xarxa Hyperledger real i, finalment, la implementació d'un entorn local amb *Hurley* com a prototip juntament amb la API amb llibreries de nodejs.

### 13.1 Competències tècniques

Quan es va presentar la proposta de TFG es van proposar quatre competències bàsiques. A continuació s'argumenta el seu assoliment:

1. Concebre sistemes, aplicacions i serveis basats en tecnologies de xarxa, tenint en compte Internet, web, comerç electrònic, multimèdia, serveis interactius i computació ubiqua: Aquesta competència s'ha assolit dintre del projecte. En aquest s'ha treballat en tecnologies d'Internet de les Coses i en el disseny i implementació d'un sistema Blockchain, que, a més, en aquest TFG s'hi ha pensat, en tot moment, des del model

de negoci des del seu disseny i des de la plasmació d'una necessitat real transformada en servei ofert per thethings.iO.

2. Dissenyar, implantar i configurar xarxes i serveis: Seguint amb l'argumentació del punt anterior, s'ha dissenyat la xarxa Hyperledger real que es portaria a producció en següents fases del projecte. A més, s'ha implementat un prototip d'aquesta xarxa utilitzant Hurley.
3. Emprar metodologies centrades en l'usuari i l'organització per al desenvolupament, l'avaluació i la gestió d'aplicacions i sistemes basats en tecnologies de la informació que assegurin l'accessibilitat, l'ergonomia i la usabilitat dels sistemes: Durant el projecte hem seguit metodologies centrades en actors i processos (extretes de la teoria de Human Computer Interaction seguint els consells del director del projecte). Aquest punt s'ha abordat àmpliament sobretot en el marc de l'EPIC.
4. Implementar i gestionar sistemes ubics: En el projecte s'ha gestionat la interacció amb dispositius IoT que han d'estar integrats en el medi físic, pel que es considera assolida la competència. En aquest TFG s'ha plasmat, sobretot, a nivell de disseny, tot i que a nivell de projecte EPIC i a thethings.iO sí que s'han arribat a fer proves amb dispositius GPS i sensors de temperatura.

A tot això, a més, es vol presentar com a annex captures de pantalla de Mansur Muaz Ekici, l'estudiant turc que ha fet una part del projecte via el projecte EPIC. Es poden veure algunes de les captures de pantalla que formaran part de l'aplicació final.

## 13.2 Altres competències

Aquest estudi, però, no només s'ha fet sota un criteri tècnic, sinó també social. S'ha posat així, en el centre, la necessitat real d'agricultors d'obtenir unes millors condicions econòmiques per a que revertís en unes millors condicions materials de vida. De la mateixa manera, s'ha apostat per projectes de comerç just en els quals la preservació del medi ambient fos un eix clau.

S'ha aprofundit també en el procés d'elaboració del cafè i de tots els passos que se segueixen a l'hora de produir-lo, des del principi fins al camp fins a l'elaboració de la tassa.

Més enllà dels objectius que s'havien plantejat, l'elaboració del TFG ha servit també per aprendre diverses tecnologies. Es començava amb un desconeixement absolut respecte la tecnologia Blockchain i gràcies a la cerca feta en el capítol de l'estat de l'art s'han pogut conèixer diverses plataformes i projectes diferents, fins aleshores desconeguts. Concretament, en el cas d'Hyperledger, ha servit per aprofundir en una Blockchain puntera ara mateix i de quines formes es pot encarar el disseny i implementació d'un model de negoci amb Fabric. Ha estat també molt enriquidor poder usar Convector, l'elecció de la qual es valora molt positivament.

## 13.3 Treball futur

A continuació es llancen algunes línies de treball futur de cara a portar a producció el projecte.

### 13.3.1 Mètriques de rendiment en casos reals

En el present treball no s'ha fet anàlisi de mètriques de rendiment de l'aplicació ja que el ser en un entorn local no tenia sentit pel cas d'ús que ens ocupa, ja que els resultats haurien estat constants i les mètriques de rendiment no tendrien sentit. Quan es porti a producció sí que serà imprescindible fer les anàlisis pertinents. A més, es podria utilitzar Blockchain Browser, que també retorna mètriques de funcionament.

### 13.3.2 Posada en producció: Kubernetes

Em l'apartat 7.3 s'ha definit la xarxa Hyperledger Fabric. Ara bé, cal entendre que en una xarxa Hyperledger s'hi forma part quan o bé es crea la xarxa de nou o bé es rep una invitació per formar part d'una ja existent. En el cas que ens ocupa qui crea la xarxa és thethings.iO qui posteriorment convidarà les organitzacions de Cafès Barcelona i Fair Trade Coffee. A la vegada, cada organització aportarà la seva pròpia infraestructura en forma de nodes. Això significa que la implementació final depèn de cada organització. Una podria optar per Kubernetes, una altra per Docker Swarm o cap de les dues.

En el cas de la implementació a thethings.iO s'optarà per Kubernetes, en bona mesura, perquè els serveis de l'empresa ja estan implementats amb aquesta tecnologia. Kubernetes és un sistema per a la gestió d'aplicacions fetes amb contenidors a través de múltiples *hosts* i proveeix mecanismes pel seu desplegament, manteniment i escalament [7]. Originalment va ser desenvolupat per Google i posteriorment es va cedir a la *Cloud Native Computing Foundation* (part de la *Linux Foundation*).

Així doncs, un cop validat el prototip es veu Kubernetes com l'opció més viable a l'hora de portar a producció el sistema dissenyat.

Amb tot, i malgrat haver optat per aquesta solució cal contemplar també una segona alternativa que facilitaria el procés d'implementació. Es tracta de *Forma*, un servei per portar a producció de forma automatitzada el projecte [6]. En aquest cas, cada organització gestiona la seva pròpia infraestructura, però Forma posa una capa per sobre per automatitzar tot el procés configurant la xarxa pràcticament de forma automàtica. Aquest, però, al ser un servei de pagament, l'elecció o no d'adoptar-lo dependrà més del departament de finances. És per això que, inicialment, s'opta per Kubernetes i fer el desplegament del servei de forma manual.

### 13.3.3 Dispositius IoT

El que s'ha dissenyat en el present TFG és capaç de rebre dades de dispositius IoT, però al ser només un prototip no s'han hagut de prendre decisions sobre els dispositius a utilitzar.

Malgrat que actualment thethings.iO treballa sobretot amb Lora o Sigfox, però les peculiaritats d'on es cultiva el cafè (tradicionalment al camp, zones altes i amb poca connectivitat a Internet) fa que les línies a explorar siguin dispositius de tipus satel·lital.



# Bibliografia

- [1] “Apache Kafka.” [Online]. Available: <http://kafka.apache.org/> 31
- [2] “Caso de éxito Quality Espresso — Vodafone grandes clientes.” [Online]. Available: <https://www.vodafone.es/c/empresas/grandes-clientes/es/grandes-clientes/casos-de-exito/quality-espresso/> 9
- [3] “Express - Node.js web application framework.” [Online]. Available: <https://expressjs.com/> 58
- [4] “fabric/mockstub.go at release-1.4 · hyperledger/fabric · GitHub.” [Online]. Available: <https://github.com/hyperledger/fabric/blob/release-1.4/core/chaincode/shim/mockstub.go> 56
- [5] “Feature: Mango Query – CouchDB.” [Online]. Available: <https://blog.couchdb.org/2016/08/03/feature-mango-query/> 55
- [6] “Forma: Multi-cloud Blockchain Infrastructure Automation - WorldSibu.” [Online]. Available: <https://worldsibu.tech/forma/> 76
- [7] “GitHub - kubernetes/kubernetes: Production-Grade Container Scheduling and Management.” [Online]. Available: <https://github.com/kubernetes/kubernetes/> 76
- [8] “GitHub - worldsibu/byzantine-browser: KHS Blockchain Browser.” [Online]. Available: <https://github.com/worldsibu/byzantine-browser> 63
- [9] “Hello Tractor Site – Hello Tractor.” [Online]. Available: <https://www.hellotractor.com/home> 9
- [10] “Hurley: Setup your Hyperledger development environment - WorldSibu.” [Online]. Available: <https://worldsibu.tech/convector/hurley-development-environment/> 46
- [11] “Hyperledger Fabric SDK for node.js.” [Online]. Available: <https://fabric-sdk-node.github.io/release-1.4/index.html> 52
- [12] “John Deere US — Products & Services Information.” [Online]. Available: <https://www.deere.com/en/> 9

- [13] “Mocha - the fun, simple, flexible JavaScript test framework.” [Online]. Available: <https://mochajs.org/> 56
- [14] “Mowi’s salmon QR codes will offer ‘searchable source of truth’ — Undercurrent News.” [Online]. Available: <https://www.undercurrentnews.com/2019/05/24/mowis-salmon-qr-codes-will-offer-searchable-source-of-truth/> 52
- [15] “Prodigio Coffee Machines — Nespresso <sup>TM</sup>.” [Online]. Available: <https://www.nespresso.com/es/en/prodigio-machines-range> 9
- [16] “Samsara — Sensores conectados a internet.” [Online]. Available: <https://www.samsara.com/es/> 9
- [17] “Sensolus — Asset Tracking Solution For Non-Powered Assets.” [Online]. Available: <https://www.sensolus.com/> 9
- [18] “thebluedots – Real time data platform for agriculture.” [Online]. Available: <http://www.thebluedots.io/> 9
- [19] “WorldSibu Documentation.” [Online]. Available: <https://docs.worldsibu.com/> 40
- [20] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich, S. Muralidharan, C. Murthy, B. Nguyen, M. Sethi, G. Singh, K. Smith, A. Sorniotti, C. Stathakopoulou, M. Vukolić, S. W. Cocco, and J. Yellick, “Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains,” 2018. [Online]. Available: <http://arxiv.org/abs/1801.10228><http://dx.doi.org/10.1145/3190508.3190538> 29, 30
- [21] Applicature, “The Latest Blockchain Supply Chain Startups – Applicature – Medium.” [Online]. Available: <https://medium.com/applicature/the-latest-blockchain-supply-chain-startups-7b0e5c07548f> 8
- [22] H. D. Benington, “Production of Large Computer Programs,” *IEEE Annals of the History of Computing*, vol. 5, no. 4, pp. 350–361, oct 1983. [Online]. Available: <http://ieeexplore.ieee.org/document/4640770/> 14
- [23] Britta Folmer, *The Craft and Science of Coffee*, elsevier ed. London: Academic Press, 2017. 1
- [24] R. Campbell, “Blockchain Tech Helps Coffee Farmers Make Fair Gains,” 2017. [Online]. Available: <https://www.ccn.com/blockchain-tech-helps-coffee-farmers-make-fair-gains> 1, 8
- [25] M. Castro, B. Liskov, *et al.*, “Practical byzantine fault tolerance,” in *OSDI*, vol. 99, 1999, pp. 173–186. 5



- 
- [26] B. O. del Estado, “Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos y Garantía de los Derechos Digitales,” p. 70, 2018. [Online]. Available: <https://www.boe.es/boe/dias/2018/12/06/pdfs/BOE-A-2018-16673.pdf> 71
- [27] European Comission, “Advancing the Internet of Things in Europe,” 2016. [Online]. Available: <https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:52016SC0110{&}from=EN> 72
- [28] —, “General Data Protection Regulation,” 2016. [Online]. Available: <https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32016R0679> 71
- [29] A. Gaihre, Y. Luo, and H. Liu, “Do Bitcoin Users Really Care About Anonymity? An Analysis of the Bitcoin Transaction Graph,” *2018 IEEE International Conference on Big Data (Big Data)*, pp. 1198–1207, 2019. 4
- [30] B. Garner, “Blockchain Agriculture Will Change Farming & Food,” 2018. [Online]. Available: <https://coincentral.com/blockchain-agriculture-change-farming-food/> 69
- [31] V. C. Hu, D. Ferraiolo, R. Kuhn, A. Schnitzer, K. Sandlin, R. Miller, and K. Scarfone, *Guide to Attribute Based Access Control (ABAC) Definition and Considerations*. National Institute of Standards and Technology Special Publication 800-162, 2014. [Online]. Available: <https://doi.org/10.6028/NIST.SP.800-162> 41
- [32] Hyperledger, “A Blockchain Platform for the Enterprise — hyperledger-fabricdocs master documentation,” 2019. [Online]. Available: <https://hyperledger-fabric.readthedocs.io/en/release-1.4/https://hyperledger-fabric.readthedocs.io/en/release-1.2/index.html{#}a-blockchain-platform-for-the-enterprisehttps://hyperledger-fabric.readthedocs.io/en/release-1.4/index.html> 28, 30, 32
- [33] Hyperledger Fabric, “Hyperledger Fabric. ABAC patterns.” [Online]. Available: <https://github.com/hyperledger/fabric/blob/release-1.4/core/chaincode/lib/cid/README.md> 41
- [34] G. Kurian, “Blockchain Technology Overview,” *The Encyclopedia of Political Science*, 2014. 3, 5, 6
- [35] W. Y. Maung Maung Thin, N. Dong, G. Bai, and J. S. Dong, “Formal Analysis of a Proof-of-Stake Blockchain,” *2018 23rd International Conference on Engineering of Complex Computer Systems (ICECCS)*, pp. 197–200, 2018. 5
- [36] Y. Mu, H. Qin, X. Liang, J. Wang, J. Li, and W. Chen, “Blockchain Based Provenance Sharing of Scientific Workflows,” *2018 IEEE International Conference on Big Data (Big Data)*, pp. 3814–3820, 2019. 3
- [37] Provenance, “Every product has a story — Provenance,” 2019. [Online]. Available: <https://www.provenance.org/> 8

- [38] N. Szabo, “Smart Contracts: Building Blocks for Digital Markets,” 1996. [Online]. Available: [http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart\\_{\\_}contracts\\_{\\_}2.html](http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart_{_}contracts_{_}2.html) 6
- [39] —, “Formalizing and securing relationships on public networks,” *First Monday*, 1997. [Online]. Available: <http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/formalize.html> 6
- [40] M. Valenta and P. Sandner, “Comparison of Ethereum, Hyperledger Fabric and Corda,” pp. 1–8, 2017. [Online]. Available: <https://medium.com/@philippsandner/comparison-of-ethereum-hyperledger-fabric-and-corda-21c1bb9442f6> 6
- [41] B. Wald and B. Brock, “Picking the Right Blockchain Learn about the different kinds of blockchains and the top technologies you should consider for your blockchain development project,” *verypossible.com*, Tech. Rep., 2017. [Online]. Available: [https://cdn2.hubspot.net/hubfs/3440604/Innovator’sGuidetoPickingtheRightBlockchain.pdf?utm\\_{\\_}campaign=Blockchain{&}utm\\_{\\_}medium=email{&}{\\_{\\_}hsenc=p2ANqtz--rwQLuiuSSYl6AJgPUtoYQdb63bGHzbnlvE7m2lS2nh{\\_{\\_}eQVfg41ivQz-c2jFiZSAIxYC7h26ACcp{\\_{\\_}sJCbG{\\_{\\_}lyuLRmMzg{&}{\\_{\\_}hsmi=5951745](https://cdn2.hubspot.net/hubfs/3440604/Innovator’sGuidetoPickingtheRightBlockchain.pdf?utm_{_}campaign=Blockchain{&}utm_{_}medium=email{&}{_{_}hsenc=p2ANqtz--rwQLuiuSSYl6AJgPUtoYQdb63bGHzbnlvE7m2lS2nh{_{_}eQVfg41ivQz-c2jFiZSAIxYC7h26ACcp{_{_}sJCbG{_{_}lyuLRmMzg{&}{_{_}hsmi=5951745) 5
- [42] Worldsibu, “Coffee Traceability Hackathon in Costa Rica – WorldSi-  
bu – Medium,” 2018. [Online]. Available: <https://medium.com/worldsibu/coffee-traceability-hackathon-in-costa-rica-72d4a75b4e64> 8
- [43] S. Yoo, S. Kim, J. Joy, and M. Gerla, “Promoting Cooperative Strategies on Proof-of-Work Blockchain,” *Proceedings of the International Joint Conference on Neural Networks*, vol. 2018-July, no. 2, pp. 1–8, 2018. 5

# Apèndix A

## Aplicació mòbil

L'estudiant que ha elaborat una de les parts del projecte EPIC, concretament el punt 3 especificat a l'Abast (capítol 2), és Mansur Muaz Ekici, de la Abdullah Gül University (AGU). A continuació es mostra una petita part del seu treball en forma de captures de pantalles de l'aplicació mòbil que ha dissenyat.

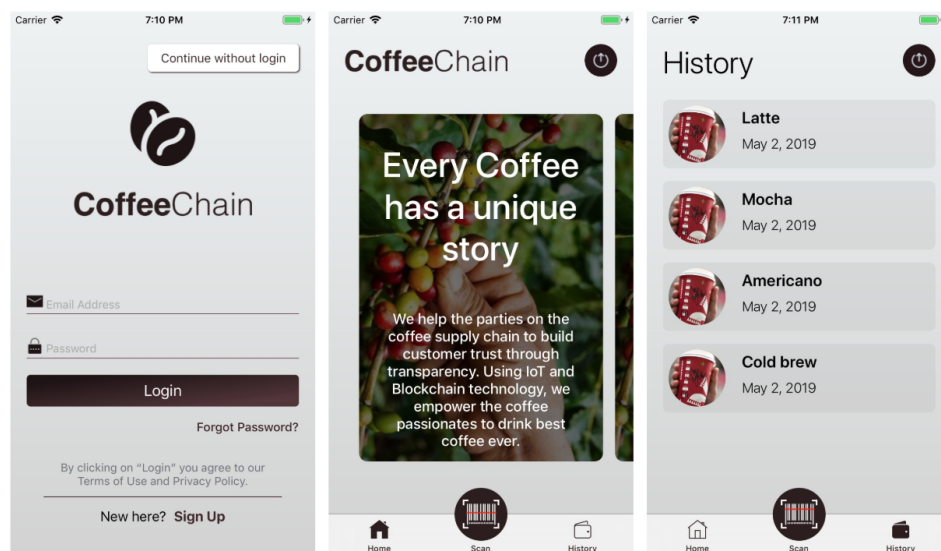


Figura A.1: Pantalla inicial i informació general

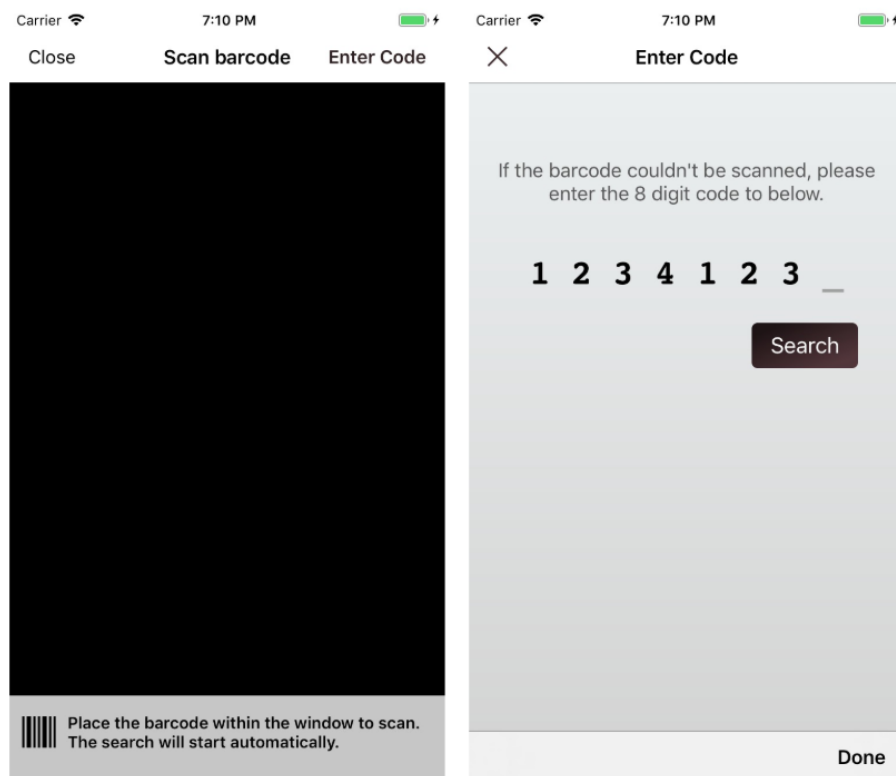


Figura A.2: Lectura d'identificadors de sacs

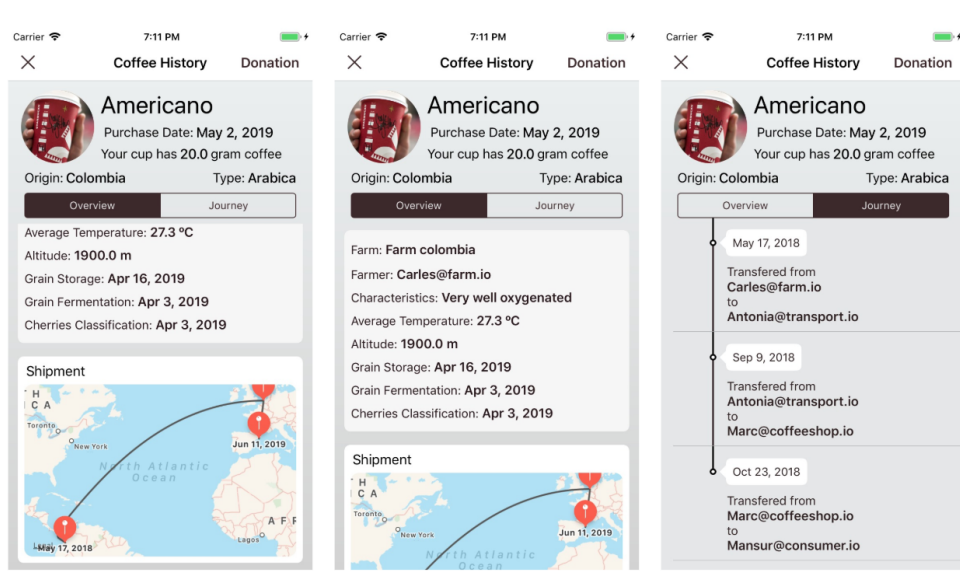


Figura A.3: Informació vària del seguiment del cafè

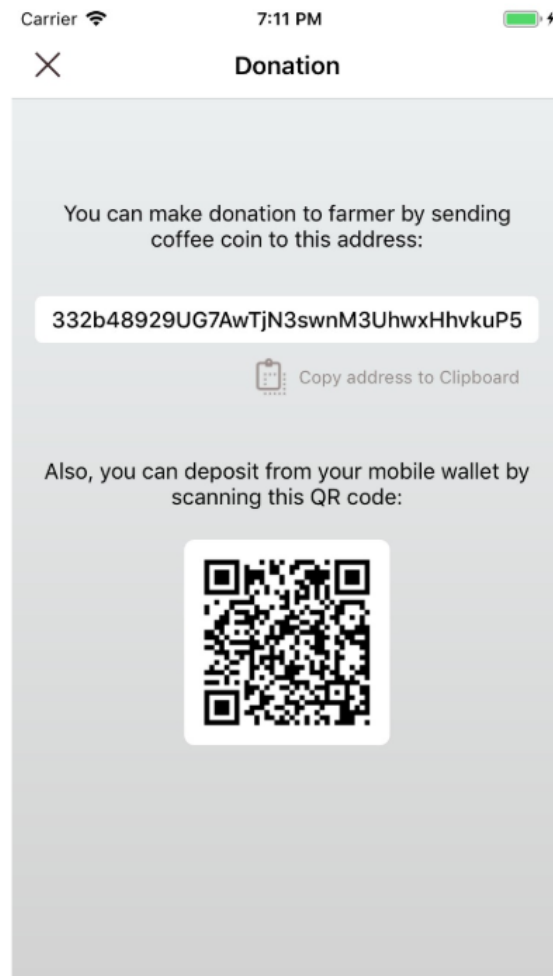


Figura A.4: Pantalla de donació a l'agricultor